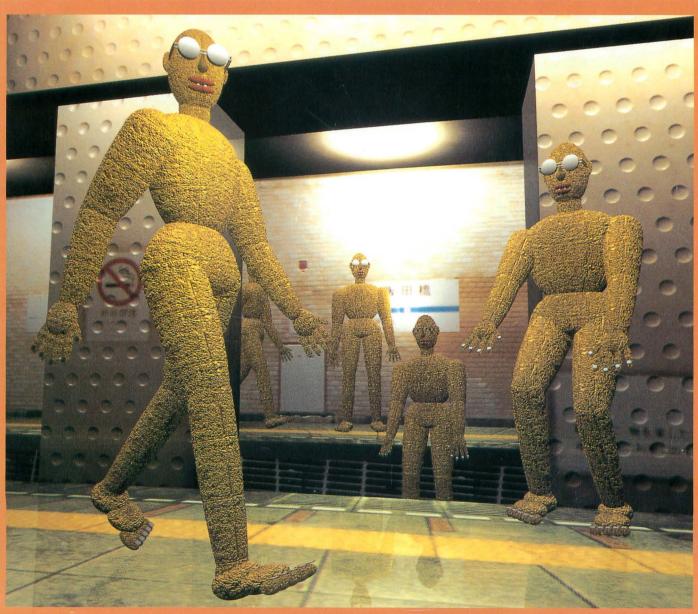
1995年1月1日発行（毎月1回1日発行）第14巻1号通巻153号 昭和58年11月2日第三種郵便物認可

特集 割り切って使うCD-ROM

CD-ROMドライブ各種紹介/CD-ROM導入の心得/PhotoCDデータの活用
BMPファイルを表示する/CD-ROMはPCMの宝庫/オーディオCDを再生する

新製品紹介 X68000XVI用アクセラレータXellent 30

1
1995

SOFT BANK オー／エックス
定価 680円

■実画面：1,024×1,024ドット、表示画：768×：512ドット

●画面は広告用に作成した、機能を説明するためのイメージ画面です。また、各種アイコンなどは、SX-WINDOW ver.3.1がもつ機能を使って作成したもので、標準装備のものとは異なるものもあります。
●本広告中の「シャーペン」で表示している文字のフォントはツァイト社の、「書体倶楽部」のフォントを使用しています。

❶「パターンエディタ」で作成したデータを背景に設定可能。

❷日本語フロントプロセッサ ASK68K ver.3.0の辞書メンテナンスがウィンドウ上で可能。

❸ESC/Page、LIPSIII、PostScriptに対応したプリンタが利用できます。

❹付属アプリケーション「シャーペン」編集例。文字ごとに文字種・文字の大きさの指定、装飾が可能。またインライン入力をサポート、イメージデータの貼りつけもOK。

❺512×512ドットの範囲内で65,536色の表示が可能。

❻「CGAウィンドウ」、65,536色（最大）のコンピュータアニメーション表示が可能。

❼異なる画像フォーマットへのコンバートが可能。

❽アイコンデータや背景データを作成する「パターンエディタ」。

❾オリジナルに作成したアイコンパターンの例。

❿Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。

フィールドが、膨らむ。

先が、ますます面白くなる。

●

未来への確かなビジョンをベースに
発展性のあるプラットホームとしてのウィンドウ環境を提供する
国産オリジナルウィンドウシステムSX-WINDOW。

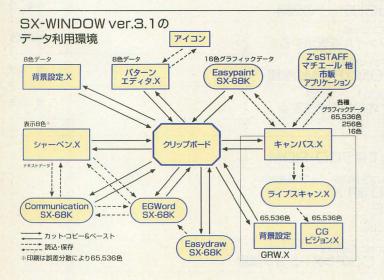
●

GUI環境や操作環境、高速化へのゆるぎない探求、
マルチメディアの統合的なハンドリング。

●

いま、より多彩なフィールドへ
そのインテリジェンスが展開を始める。

●

次のステージが見えてくる。

●インライン入力のサポート：ASK68K Ver.3.0を利用したインライン入力をSX-WINDOWで実行可能。またシャーペン.Xをワープロとして利用できるよう、さまざまな機能が付加されています。

●コンソールをサポート：Human68kやX-BASICのコマンドをSX-WINDOWアプリケーションと同時にタイムシェアリングで実行できます。
（グラフィックを利用したものなど、SX-WINDOWと処理が重複するものは実行できません。）

●多彩なプリンタに対応：さまざまなSX-WINDOWアプリケーションで利用できるページプリンタドライバを標準装備。ESC/Page、LIPS III、PostScriptに対応したプリンタが利用できます。

今も、先も楽しめる。
いつも新展開の予感、SX-WINDOWのニューバージョン。

SX-WINDOW ver.3.1

「SX-WINDOW ver3.1システムキット」CZ-296SS（130mmFD）/CZ-296SSC（90mmFD） 標準価格22,800円（税別）

EXEディスク2プレゼント

シャーペンカスタマイズコンテストの力作や、新作SX-WINDOWソフトウェア情報などを満載のディスク情報誌「EXEディスク2」をプレゼントいたします。

●官製ハガキに住所、氏名、EXE会員番号と90㎜（3.5型）/120㎜（5.25型）の種別を明記の上、お申込み下さい。また、これからEXEクラブへ入会される方は、商品同梱のEXEクラブ入会申込書に「EXEディスク2希望」と明記の上、ご投函下さい。

応募/問合せ先

〒545 大阪市阿倍野区長池町22-22 シャープ株式会社電子機器事業本部システム機器営業部EXEクラブ事務局EXEディスク2係宛
（TEL 06-621-1221大代表）

申込締切

平成6年12月末日消印有効

●EXEクラブに入っておられない方は、ソフトベンダー「TAKERU」での購入が可能です。
（平成6年11月1日より2ヶ月間、予価200円）

シャープ株式会社

資料請求券 ペリフェラル Oh!X 1体

特集　割り切って使うCD-ROM

パックランド

上海　万里の長城

魔法大作戦

新製品紹介　Xellent 30

(で)のショートプロぱーてぃ

Oh!X

CONT

●特集

〈スタッフ〉

●編集長／前田　徹　●副編集長／植木章夫　●編集／山田純二　豊浦史子　高橋恒行　●協力／有田隆也　中森　章　林　一樹　吉田幸一　華門真人　朝倉祐二　大和　哲　村田敏幸　丹　明彦　三沢和彦　長沢淳博　司馬　護　清瀬栄介　石上達也　柴田　淳　瀧　康史　横内威至　進藤慶到　●カメラ／杉山和美　●イラスト／山田晴久　江口響子　高橋哲史　川原由唯　●アートディレクター／島村勝頼　●レイアウト／元木昌子　ADGREEN　●校正／グループごじら

表紙絵：塚田　哲也

E　N　T　S

●THE SOFTOUCH

●シリーズ全機種共通システム

●連載/紹介/講座/プログラム

1995 JAN.
1

■広告目次

ビデオグラフィックスの
世界へ。
■お問い合わせは… シャープ株式会社
電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
資料請求券
X68030
Oh!X

1,677万色対応、ビデオ映像を高画質・高速取り込み

テレビやビデオ、ビデオディスクなどの映像をX68シリーズやMacシリーズ※1の動画・静止画データとして高速取り込みが可能、いわば“ビデオスキャナ”とでも呼びたいビデオ入力ユニットです。1,677万色対応、最大640×480ドットの高解像度※2。動画・静止画の手軽なハンドリングが、新たなグラフィックシーンを創造します。

※1 MacintoshはIIシリーズ以降の機種に対応、ディスプレイ解像度が640×480ドットの場合、取り込み可能な範囲は、160×120ドット、320×240ドットのサイズになります。
※2 X68030/X68000シリーズでは、1,677万色はデータ作成のみに対応。表示は最大65,536色、解像度は512×512ドット。また、Macintoshは機種により表示色数が異なります。

アプリケーションツール「ライブスキャン」を標準装備

動画や静止画を簡単に保存できるアプリケーションソフト「ライブスキャン」※を標準装備。取り込んでいる映像を表示したり、残したいシーンを簡単に静止画保存したり、手軽な動画・静止画ハンドリングでパソコンの可能性をさらに広げます。X68030/X68000シリーズ用SX-WINDOW対応版とMacintoshシリーズ用QuickTime対応版の2種類を同梱しています。

※SX-WINDOW版はバージョン3.0以降(メモリー4MB以上)、QuickTime版はMacintosh漢字Talk7リリース7.1以上のシステムとQuickTime1.5以上(メモリー8MB以上)が必要です。

1,677万色対応の高速映像取り込み、動画・静止画の手軽なハンドリングが、新たなマルチメディアシーンを創造する。

■SCSIインターフェイス採用:パソコンの専用I/Oスロットを使わずに接続可能になり、汎用化を実現しました。またSCSI-2(FAST)インターフェイスの採用により、データ転送速度の高速化を図っています。X68030/X68000シリーズでは、SCSI-2(FAST)対応のハードディスクを接続することにより、パソコン本体を経由しないで、ハードディスクに直接、動画データをテンポラリデータとして記録することが可能です。パソコン本体のハードディスクへは、記録終了後に、テンポラリデータを変換し動画データとして保存できます。

※CZ-600C/601C/611C/602C/612C/652C/662C/603C/613C/653C/663Cに接続する場合は別売のSCSIインターフェイスボードCZ-6BS1ならびにSCSI変換ケーブルCZ-6CS1が必要です。※CZ-604C/623C/634C/644Cに接続する場合は、別売のSCSI変換ケーブルCZ-6CS1が必要です。
※Macintosh Power Bookシリーズに接続する場合は別売のSCSIケーブルなどが必要です。詳しくはMacintosh Power Bookシリーズの取扱説明書をご覧ください。

■高機能MPUを搭載:クロック周波数25MHzの32ビットMPU/MC68EC020を搭載、高速処理やパソコン本体の負担の軽減を実現します。

●MacはMacintoshの略称です。●Macintosh、MacintoshIIは、米国アップルコンピュータ社の登録商標です。●Power Bookは米国アップルコンピュータ社の商標です。●漢字Talk7はアップルコンピュータジャパン社の商標です。●QuickTimeは、米国アップルコンピュータ社の商標です。●価格には、消費税及び配送・設置・付帯工事費、使用済み商品の引き取り費等は含まれておりません。

ビデオ入力ユニット

CZ-6VS1

標準価格178,000円(税別)

●拡大読み取り時、細かい部分でも忠実に再現。
2400dpi※1やデジタルズーム機能が高品位を守ります。

●35ミリフイルムも透過原稿読み取りユニットを使用して読み取り可能。

高速処理を実現。スピーディに作業できます。

A4、300dpiならカラー約13秒※2、モノクロ約1秒※2でこのクラス最高の※3高速読み取りが可能です。大きな画像データを高速転送できるSCSI-Ⅱにも対応。また、最大A4/リーガルサイズ(216.4×355.6mm)までの原稿を読み取りできます。

高解像・高品位。美しさが際立ちます。

基本解像度600dpi、疑似解像度2400dpi※1の高解像度読み取りで微細な点や線を鮮明に再現します。縮小・拡大は30〜2400dpiの範囲で設定可能です。また、約1677万色で原画に忠実なリアルな色合いを再現します。

●シャープ独自の技術「デジタルズーム」搭載により繊細な線やズーム画像も忠実に再現。また「ワンウェイスキャン方式」を採用し、凹凸のある原稿も鮮明に読み取りできます。

通常の拡大時
(当社従来機)

デジタルズーム
(JX-330シリーズ)

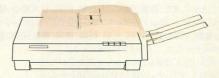
透過原稿読み取りユニットとADFを同時装着できます。

透過原稿読み取りユニットは、35mm(ネガまたはポジ)フィルムからレントゲン写真まで各種透過原稿※4に対応。基本解像度600dpi/1200dpiの2種類をご用意しました。また最大50枚までの原稿を自動送りできるADFも同時装着できます。※5

X68000対応カラーイメージスキャナ

JX-330X

透過原稿読み取りユニット(オプション)
JX-3F6 標準価格 98,000円(税別)
JX-3F12 標準価格138,000円(税別)

カラーイメージスキャナ
JX-330X 標準価格178,000円(税別)

ADF[原稿自動送り装置](オプション)
JX-AF3 標準価格 58,000円(税別)

使いやすい高機能画像入力ソフトを標準装備〈JX-330X〉
●Scanner Tool/S(画像入力ソフト)、対応フォーマット形式:ZIM、PIX、GL3、PIC、GLX、GLM

※1 2400dpiは当社独自手法による疑似解像度です。※2 読み取り開始から読み取り終了までの動作時間。ただし初期動作およびデータ転送時間を除く。(室温25℃)※3 クラスとは、A4フラットベットクラスのこと。'94年11月現在。
※4 読み取り可能なサイズは機種によって異なります。※5 ご使用になるアプリケーションにより対応が異なります。

■消費税及び配送・設置・付帯工事費・使用済み商品の引き取り費等は、標準価格には含まれておりません。

IMAGICA

IMAGICA TECHNOSYSTEMS

RAIZING

魔法大作戦

パワーアップして撃ちまくれ。敵は悪の王、ゴブリガン。

ゴブリガン王率いる亜人国家、ゴブリガン帝国の来襲により、
平和な国家は危機に瀕していた。
事態を重くみた国王コードウェナーは、ゴブリガン王の首に
多額の賞金をかけ、王国を危機から救い出す勇者を募った。
名乗りをあげたのは、
強腕戦士ガイン、魔法使いチッタ、侍竜ミヤモト、呪術師ボーンナムの
4人の勇敢な戦士たち。いずれ劣らぬ戦いのプロであった——。

- アミューズメントパークで好評を博した、縦スクロール・シューティングゲームの完全移植版。
- 7つのステージで繰り広げられる白熱のシューティング。
- プレイヤーを飽きさせない多彩な風景、個性的なボスキャラたち。
- 派手なパワーアップでシューティングを堪能。
- よりアーケードに近い感覚でゲームを楽しめる縦画面モードを用意。

▲強腕戦士ガイン

▲魔法使いチッタ

▲侍竜ミヤモト

▲呪術師ボーンナム

X68000シリーズ	FM-TOWNS
価　格：9,800円	価　格：9,800円
発売予定日：'94年12月	発売予定日：'95年春
メディア：5インチFD	メディア：CD-ROM
※要2M RAM	※要2M RAM

ELECTRONIC ARTS™

エレクトロニック・アーツ・ビクター株式会社
〒150 東京都渋谷区神宮前2-4-12　フルークス外苑
商品に関するお問い合わせ：カスタマーサポート係 TEL.03-5410-3100（受付時間13:00～16:00/月～金）

通信販売：当社の製品をお近くのパソコンショップでお買い求めになれない場合、通信販売もご利用いただけます。商品名、機種名、住所、氏名、電話番号を明記の上、左記住所まで定価プラス3%消費税分を現金書留にて営業部通信販売係宛にお送りください。（送料当社負担）

ハイパーアクション プロレスカードゲーム

レッスルエンジェルス SPECIAL

セクシーでパワフルな女子プロを制覇しろ！

18禁版

カードバトルにプロレスを融合させた、「レッスルエンジェルス」シリーズ。いよいよ最大のヒット作「レッスルエンジェルススペシャル」が登場です。さまざまなイベントの選択によって運命が変わる、マルチシナリオ・マルチエンディング。プロレス技数、カテゴリーが増加して、レスラーの個性もパワーアップ。そして、「恐怖の水着はぎデスマッチ」もパワーアップして復活！18禁だから、そのセクシー度はもうケタ違い！待望のX68000移植完成！明日のトップイベンターを目指すのだ！

機能アップ！

- ●オリジナルオープニングを収録
- ●画面のレイアウトを変更
- ●エキジビションモードグラフィック描き直し
- ●256色モードと16色モードを搭載
- ●サウンドも明るめに変更
- ●AD-PCMによる効果音
- ●ディスクアクセスを最少に抑える設計

このソフトは、全国のパソコンショップで、パッケージ版で販売いたします。TAKERUでは販売致しません。TAKERU事務局では通信販売はいたしませんので、悪しからずご了承下さい。

対応機種：X68000/X68030
要メモリ2Mバイト
（ハードディスク対応）

制作：グレイト

￥8,800（税別）

三國志

知力の極限に挑む、君主、武将、軍師の膨大なデータ。小説よりリアルと、名作の誉れ高い中国統一ゲーム。この歴史的な傑作シリーズはどのようにして始まったのか?SLGファンなら絶対に見逃せない!!

制作/光栄
対応機種/X68000（30不可）　**¥5,200**

太閤立志伝

裸一貫の足軽頭から身を興し、関白にまで登り詰めた男・木下藤吉郎(豊臣秀吉)。草履を温めたエピソード・奇跡の墨俣一夜城など、数々の逸話を持つ男の一生を再現する、リコエイションゲームの傑作です。

制作/光栄
対応機種/X68000（30不可）　**¥3,400**

ファランクス

デカキャラ・派手め演出の横スクロールオアワーシューティング。拡大・回転・縮小・多関節・半透明・ラスタースクロール・MIDIと、各種要素がいっぱい詰まってます。

制作／ズーム
対応機種／X68000（30不可）　**¥2,500**

三國志 II

登場人物350余名、最大11人まで同時プレイ可能、6編のマルチシナリオ方式、埋伏の毒・駈虎呑狼等のユニークな計略要素導入、さらに深みを増した外交・HEX戦など、まさに名作!カシオペアの向谷　実のBGMも話題に。

制作/光栄
対応機種/X68000（30不可）　**¥4,900**

蒼き狼と白き牝鹿　元朝秘史

光栄歴史三部作の一角を成す、草原の英雄チンギス・ハーン。稀代のスケールと空前絶後の迫力で、一代帝国を築き上げた男の豪快な一生を見事に再現!熱いシミュレーションの傑作です。

制作/光栄
対応機種/X68000（30不可）　**¥3,400**

A列車で行こうII

かの「A列車」シリーズの第2弾。パズル的要素がアツクなる!鉄道会社社長の立場で、線路の敷設・撤去を行い、ワールドワイドにマップを発展させていこう。

制作／アートディンク
対応機種／X68000（30不可）　**¥3,800**

大航海時代

リコエイションゲームシリーズの傑作。毎回違った展開が楽しめるイベントジェネレーティングシステム。帆船の特徴が活かされたHEX戦。失われたロマンを求めて、冒険者たちの航海の旅が始まる。

制作/光栄
対応機種/X68000（30可）　**¥3,400**

ロイヤルブラッド

新シリーズ「イマジネイションゲーム」のデビュー作。イシュメリアという架空の島国を舞台にした、幻想世界のシミュレーションゲームだ。あなたは独立貴族のひとりとなり、領主達が持っている6つの宝石を集め、イシュメリアの新王となれ!

制作/光栄
対応機種/X68000（30可）　**¥2,700**

A III（A列車で行こう3）

さらにワイドに、さらに完成度の増した、世界レベルヒットの第3弾。世にA.IIIブームを巻き起こしたことで、記憶に新しい超有名作、ついに文庫に登場！

制作／アートディンク
対応機種／X68000（30可）　**¥3,800**

維新の嵐

坂本龍馬が、西郷隆盛が、吉田松蔭が日本を憂い、改革を目指して奮い立つ!幕末の志士の個性を際だたせる緻密なパラメータ。出会いの楽しさ、駆け引きを楽しむ新システム。強力な機能で、維新を操れ!

制作/光栄
対応機種/X68000（30不可）　**¥3,400**

ヨーロッパ戦線

戦乱のヨーロッパ。砂塵の彼方から迫り来る黒い車体は、敵か味方か?次々に飛び込んでくる情報、時事刻々と変わる戦局。多彩な兵器やユニット、人間的要素を重視した各種パラメータ。WWIIシリーズ第2弾。勝利の旗を手に入れろ!

制作/光栄
対応機種/X68000（30可）　**¥4,500**

栄冠は君に

高校野球シミュレーションシリーズの、記念すべき第1作。全国制覇を達成するには、3990校の頂点に立たなければならない。感動の優勝セレモニーを、果たして見ることが出来るか！？

制作／アートディンク
対応機種／X68000　**¥3,800**

信長の野望　戦国群雄伝

400余名の群雄が割拠する下剋上の乱世。配下の羽柴秀吉、柴田勝家を個性豊かな武将たちを、思いのままに操って、戦雲たなびく戦場へ、天下分け目の決戦に臨む！光栄の代表作「信長の野望」シリーズの傑作！

制作/光栄
対応機種/X68000（30可）　**¥3,400**

大戦略 III '90

90年代にふさわしくパワーアップされた「大戦略」シリーズ。戦略思考ルーチン、ゲームスピード、コマンド体系、リアルタイムオペレーションなど大幅革新された作品です。

制作／システムソフト
対応機種／ X68000　**¥2,500**

ルーンワース「黒衣の貴公子」

ハイドライドシリーズに続く、新ARPGシリーズ第1弾。綿密に構築された世界「ルーンワース」を舞台に、極めて自由度の高いゲームシステムの中で、興奮の冒険が始まります。

制作／T&Eソフト
対応機種／X68000　**￥700**

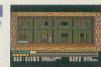
伊忍道　打倒信長

1つのゲームでSLG とRPG、2つのジャンルが楽しめるリコエイションゲームの第3弾。特にRPGの要素が濃い、異色傑作だ!意志を持ったキャラクターが目的に向かって行動を展開。敵を倒して腕を上げ、技を磨いて信長を倒せ!

制作/光栄
対応機種/X68000（30不可）　**¥3,400**

ジェノサイド 2

あのズームのゲームがついに名作文庫に登場！特大キャラとハデハデな演出で、68ユーザーのどぎもを抜いた名作アクションゲームだ。MIDIにも対応しているぞ。

制作/ズーム
対応機種/X68000（30不可）　**¥2,500**

イース III（ワンダラーズフロムイース）

よりアクション性を増した、これまた、大人気を博したアクション・ロールプレイング。アドルの最後の冒険物語でした。攻撃方法もいっそう多彩になって、時間を感じさせない逸品です。

制作／日本ファルコム
対応機種／X68000（30不可）　**¥2,000**

パソコンソフト自動販売機
TAKERU事務局
〒467 名古屋市瑞穂区苗代町2番1号
ブラザー技術開発センタービル2F
TEL(052)824-2493（受付時間：月～金　13:00～18:00）

営業所
東京営業所 (03) 5443-4967
大阪営業所 (06) 258-3024

通信販売 1994年4月1日より、送料／手数料が有料になりました。

ソフト名、機種名、メディアのサイズ、住所、氏名、電話番号を明記の上TAKERU事務局まで現金書留でお申し込みください。送料／手数料は、1回のお申し込み総金額が5,000円以上の方は無料。4,900円までの方は500円をいただきます。4,900円までの方は現金500円をプラスしてお申し込みください。誠に勝手ながら、皆様のご理解とご協力の程、お願い申し上げます。

MIKI

スーパーリアル麻雀 PⅡ&PⅢ ファンブック

KASUMI

しょう子・香澄・未来……
あの3人にまた会える!!

「……むかし、むかしあるところに麻雀のたいそう強い娘が3人いた。その娘たちの名前は、しょう子、香澄、未来。それはそれはかわいくて愛くるしい娘たちだった。多くの人々が、100円玉を握りしめ、その娘たちに会いに行ったという。しかし、彼女たちの本当の姿を目にした者は少なかった。そして、いつしか人々は娘たちを"麻雀の女神"と呼ぶようになった……」

(スーパーリアル麻雀神話・序章より抜粋)

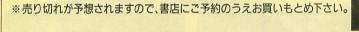
今や伝説・神話となった名作PⅡPⅢ初の公式ファンブック。幻の原画・コンテや描き下ろしセル画、コミックなど貴重な資料が満載の完全保存版!

SHOWKO

豪華3大付録

① オリジナルピンナップ
② パラパラアニメ写真
③ 着せ替え紙人形

PⅡ&PⅢ 初の公式ブック登場!!

※売り切れが予想されますので、書店にご予約のうえお買いもとめ下さい。

好評発売中
定価2,000円

●・・大好評発売中・・●

スーパーリアル麻雀PⅣ 原画&設定資料集

- ●定価2,000円
- ●A4判

豪華ゲスト描き下ろしイラスト満載

A3描き下ろしポスターつき

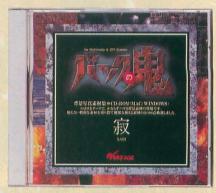
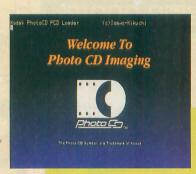
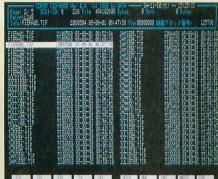
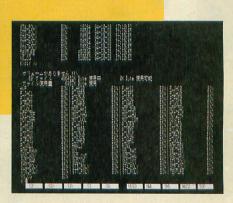
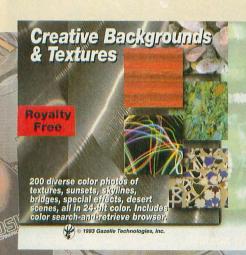
SOFT BANK

■定価は税込みです

ソフトバンク株式会社/出版事業部　販売局:TEL 03-5642-8101

割り切って使う CD-ROM

最近はCD-ROMのかたちでさまざまなタイトルが発表されています
国産のもの，外国産のもの，パソコンのもの，ゲーム機のもの
それらのなかからX68000で使えそうなデータを見てみました

もっとも入手しやすいデータといえるBMPファイル。SX上から背景に表示するビュアを作成してみた。ほとんどが256色データなのが残念。キックの鬼なんて知らない世代も多いと思うが……。

コダックPhotoCDは今後の大規模データで標準となるべきデータ形式だ。Photoデータ集の作成は非常に簡単（当たり前）で，たくさんのデータ集が発表されてきている。ローダはディザつきなので画質は○。右はDiver'sSelectionから。元が写真だから当然横位置のもある。

フリーソフトウェアのAVI_PLAYを使って，WindowsのAVIファイル（動画形式）をX68000上で再現しているところ。なんでも再生できるわけではないが，利用価値は十分に高いツールだといえるだろう。いちばん右はCDROM.X。よくできたツールだ。

左から，計測技研のMacFileTransfer.x，CD2PCM.Xの実行画面，そしておまけとしてSATURN用バーチャファイターのCD-ROMをドライブに入れてDIRを取ってみたものだ。

テクスチャー素材を集めたCD-ROMは実にたくさん発売されている。マッピング素材を求めている人が多いのか？

ビーナスの美女 VENUS®
スイーム・スーツ CD-ROM I
流行の水着に身を包んだ，美しく，官能的なモデル達の写真が425枚！

これが，CDROM.Xの作者が開発のきっかけのなったという"WOMEN IN VENUS"の日本版だ。深い意味はない。

CS-CD301X
コパル

おそらく「黒いFilo」でX68000ユーザーにはデザインが大事なことを実感したものと思われます。色が黒いことは当然として，横幅と奥行きが縦型X68000と同じであること（154×270mm），電源スイッチが前面に配置されていることなど，実にポイントを押さえた作りになっています。

ちょっと気になるのは，電源ケーブルが短いこと，イジェクト時にトレイ（というかドライブの本体）が少ししか飛び出さないということでしょうか。あとは手で引っ張り出して入れ替える必要があります。しかし考えようによっては，このような仕様のほうが使いやすいのかもしれません。このドライブは形状からして，X68000本体の下に置いて使う局面が多くなるものと思われます。とすれば，ケーブルやマウスの密集しているところでトレイがフルオープンされるのも困ることがあるでしょう。

欠点は，それでもなお置き場に困るということでしょうか。本体にジャストフィットの大きさなので，やはり本体の下に置かなければメリットにはなりません。

しかし，当然予想されることですが，キーボードやマウスのケーブルは邪魔になります。使用環境にもよりますが，前面の扉をまるごと引き出すので，それらしい場所にマウスマットを置けないというのもちょっと困ります。せっかくの美点を無駄にするには忍びないので少し考えてみましょう。美しく使うためには，

1) キーボード延長ケーブルで本体前面を回避する
2) マウスはキーボードに接続する
3) マウスマットは使わない

ということになります。

この製品は東芝ドライブを使用しており，X68000で使ううえでのソフトウェア的な問題はほとんどありません。ドライバソフト（計測技研のもの）とケーブル（SCSI2用），ターミネータが付属しており，必要十分のパッケージングとなっています。かなりお買い得といえるでしょう。

接続状況にもよるのでしょうが，ドライブの電源を入れていないと本体が正しく起動しないこと（普通はたいてい大丈夫なのだが），CD-ROMを入れっぱなしでSX-WINDOWなどを起動するとメディアを認識しないことなどの問題点もあります。

CS-CD301X　29,800円（税別）
コパル　☎03（3965）1161

CDS-E
・
満開製作所

この製品は，ドライブ自体はメルコの製品なのですが，満開製作所が計測技研のドライバとセットでX68000用として通信販売しているものです。

CDS-E自体は以前，瀧氏の紹介記事にもあったとおり，低価格で必要な機能はすべて揃っているという手頃感のあるドライブです。

CDS-Eは内部にソニー製ユニットを採用していますので，X68000で使用してもソフト的に問題が出ることはほとんどありません。ハード的に見ても非常に安定しています。全体的な印象でいって，素直に動くドライブということであれば今回紹介した製品群のなかでも一番でしょう。

計測技研のCD-ROMドライバver.2.0の旧版に付属していたツールでは，なぜかCDプレイヤーが動作しかったのですが最新版では改善されましたし，MacFileTransfer.XでMacintosh用のCD-ROMをアクセスすることもできますし，フリーソフトウェアのCD2PCM.Xを使用すれば音楽CDデータの読み込みもできます。ただし，変換の速度はかなり遅いものです（CS-CD301Xの数倍かかる）。

ドライブ自体の仕様は2.4倍速となっていますが，普通に使うと2倍速のままです。残念ながら，ソニー，メルコともコマンド非公開ということで2.4倍速対応は断念されたようです。

なお，今回紹介した製品はすべてトレイ式のメディア出し入れが可能なものですが，動作としてはこの機種がもっともオーソドックスです。もちろんソフトウェアでフルオープン＆ローディングが可能となっています。

ちなみに，満開製作所のX68000用CD-ROMセットのラインアップとしては，この製品のほかにも緑電子製の4倍速ドライブ（東芝製ユニットを採用）によるものも予定されています。これは現状のCD-ROMドライブ製品群のなかでシーク速度，転送速度，信頼性などを総合して考えてみた場合，最高レベルの性能を持つCD-ROMドライブだといえます（ただし，この製品はトレイ式ではなく，キャディ式になっているので注意）。

どうしても最高級品がいいんだという人はそちらをどうぞ。

CDS-E　29,800円（税別）
満開製作所　☎03（3554）7441

SCD-200

ロジテック

現在市場に出ているCD-ROMドライブで，もっとも低価格なのが，このロジテックSCD-200です。この製品にはドライバもケーブルもターミネータも付属していません。まあ，X68000で使うには他機種用のドライバやインタフェイスボード同梱で値段が上がるよりはよいのかもしれませんが。

定価合計は本体19,800円＋ドライバ6,800円＋ケーブル約5,000円でだいたい31,600円。ちょっと高めに感じる人もいるかもしれません。今回取り上げた別の2つがいずれも直販特価29,800円で値引きが期待できないのに対し，こちらは実売で25,000〜27,000円程度になることが予想されます。普通のパソコンショップで入手できるというのもメリットのひとつでしょう。

X68000で使うにはそれなりの問題点もあります。まず，この製品に実装されているNECドライブは計測技研のドライバとツールではうまく動作しない部分があります。さらにCD-ROMドライバのサンプルでついてくるSX-WINDOW用のCDプレイヤーソフトが動作しません。まあ音楽用CDとは別物だと思えば特に問題はないでしょう。

次に，SX-WINDOW上でMacintoshのファイルが読めません。世の中にはMacintoshのようなひねくれたCD-ROM以外にもちゃんとしたCD-ROM（Windows用など）が増えてきましたし，コマンドライン上ではCDROM.Xで問題なくアクセスできますから，SX-WINDOW上では使えないもんだと割り切れば特に問題はないでしょう。

また，NECドライブでは音楽CDのオーディオデータを読み込むことができません。これはあきらめましょう。

最後にソフトウェアでイジェクトできません。イジェクト作業をしているようにも見えるのですが，トレイは出てこないようです（これもドライバ側の問題と思われます)。これは指でイジェクトボタンを押してやればよいだけの話ですね。

ハード的には，SCSI IDが6番に固定されていますが，そうそうまずい事態になることはないでしょう。

なんか，思いきり都合の悪いドライブにも見えますが，CD-ROMドライブというのはデータ流用という見地に立てば，最低限必要な機能だけでも十分な作業ができるものなのです。

SCD-200 19,800円（税別）
ロジテック ☎0265(74)1455

Oh!X Graphic Gallery

DōGA CGアニメーション講座

先月号の特別付録「XL/Imageお試し版」はいかがでしたか。製品版もすでに発売されています。そこで，今月号のDōGA CGアニメーション講座では，このXL/ImageをCGAシステムのレンダラとして使ってみます。CGAシステムのRENDによる作画と比べてみてください。

四角錐の作画例（本文76ページ）

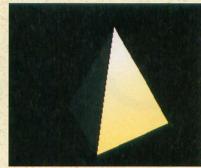



写真1　XL/Imageによる

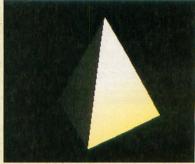



写真2　CGAシステムのRENDによる

コックピット周辺にマッピングを施した宇宙戦闘機の作画例（本文76ページ）

マッピングデータ

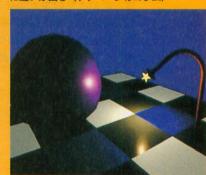
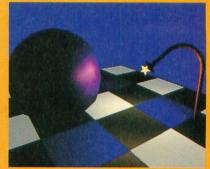
写真3　XL/Imageによる

写真4　CGAシステムのRENDによる

XL/ImageでCGAシステムのRENDと同様の表現を目指してみた（本文80ページ）

写真5　XL/Imageによる

写真6　CGAシステムのRENDによる

XL/ImageとCGAシステムではシェーディングの方法が異なるため，ハイライトのつき方に違いが出る（77ページのコラム）

XL/Image（フォンシェーディング）

CGAシステム（グーローシェーディング）

[第44回]
完ぺきの国産テレビ

江口　響子

響子inCGわ〜るど

東京，目黒の祖母の住む家までは，二子玉川の実家から小1時間だった。母に連れられて，よく遊びに行った。

戦前からの洋館造りの家の片隅に，ひっそりと置かれていたものがあった。

古い型の白黒テレビである。祖母は，目が悪くならないようにと，青く透明なプラスチックを画面の上にかけていた。

そのテレビは，機能を実現するのに精一杯で，デザインは二の次という感じがした。しかし，ずんぐりとした本体と丸みをおびた画面は，昔のSF映画やテレビアニメに出てくるロボット……たとえば鉄人28号のように，素朴な愛敬があったように思う。

『輸入品に劣らぬ性能　完ぺきの国産テレビ』

1953年に，国産第1号のテレビが発売された。これは，そのときのキャッチコピーである。価格は175,000円。当時の大卒の初任給が1万円前後だから，現在でいうと350万円ぐらいであろうか。近頃のハイビジョンテレビなど問題にならないくらいの高額商品だ。

同じ年に，NHKのテレビ放送が始まった。放送時間は1日に4時間半のみ，最初の受信契約者は全国で866人だった。

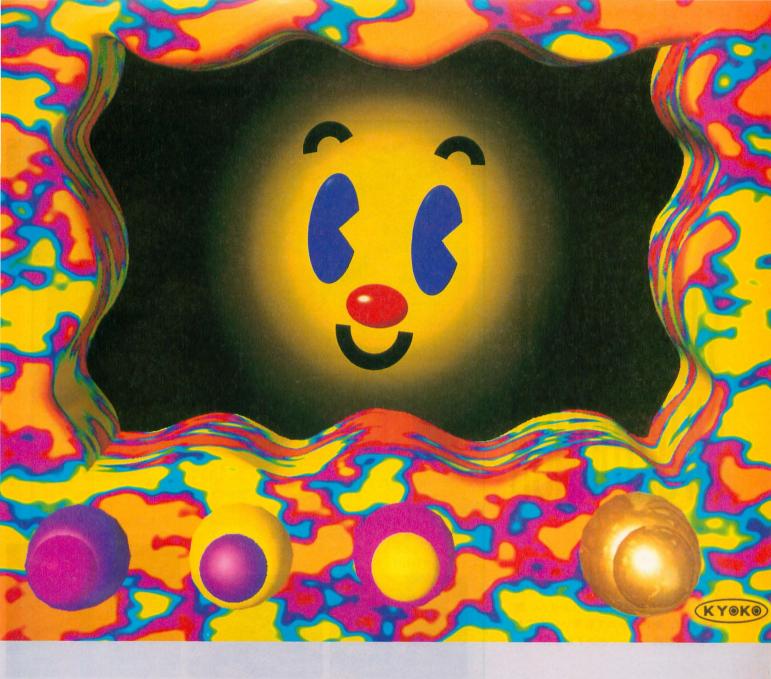
当時の気分を，父母やその同年代の人たちに聞くと，戦争の傷あとはまだ覚めやらぬものの，まったく新しい自由な物の考え方や生き方を知って，未来が開けているようだった……ということである。朝鮮戦争による特需，それに続く神武景気，岩戸景気による経済的な裏うちもあったのだろう。そうした中で，未来への夢と技術が結実し，さまざまなアイデアが商品化されていった。

1953年・国産第1号の噴流式洗濯機『欧米で一番の人気』三洋電機。28,500円。

1955年・自動式電気釜『眠っている間に，ご飯がたける』東京芝浦電機（現東芝）。3,200円。

1958年・インスタントラーメン「チキンラーメン」『この手軽さに，女房，大いに喜ぶ』日清食品。35円。

1961年・大衆車「パブリカ」（697cc）『これ以下ではムリ，これ以上では無駄』トヨタ自動車工業。389,000円。

どれも現在の生活に溶け込んでいるものばかりである。

振り返って現在。未来への希望や夢を技術力で結晶させるエネルギー，本当に必要で新しいものを作り上げる時代のエネルギーは，どんな形で現れているのだろうか。それとも，もうどこかへいってしまったのだろうか。

『輸入品に劣らぬ性能　完ぺきの国産テレビ』

国産第１号のテレビを作ったのは，早川電気工業，現在のシャープである。そういえばX68000には『感性を光らせる』というキャッチコピーがあったなあ～。ぼんやりと目の前にあるマシンとディスプレイテレビを眺めていると，新しいものが好きだった祖母と白黒テレビのことが思い出されるのだった。

今回のCGデータ

1280×1024ピクセル
1670万色フルカラーを4×5ポジで出力
作成手順
X68000SUPERでキャラクターの画像をCSGとファイヤーボールで作成。
使用ソフトは，XL/Image
MacintoshQuadra840AVでテレビの画像をポリゴンで作成（使用ソフトは，STRATA STUDIO Pro）
XIN/XOUTIII(電机本舗)で，キャラクターのRGBファイルをX68000からMacintoshに転送したのち，Photoshop2.5Jでコラージュ

SOFTWARE INFORMATION

12月には新作ソフトがいくつか続けて発売されました。夜の長い冬は，パソコンにはうってつけの季節かもしれません。熱中すれば，寒さも忘れて暖房いらずでとっても経済的……ってホントかな!?

VIEW POINT

当初の予定がやや延期されて，年が明けてから発売されることになった。期待しているのになかなか出てこない！　といわれるかもしれないが，もう少し皆さんには待ってもらおう。開発は着々と進行しているようだから，安心するように。

この「VIEW POINT」というゲームは，視点がクオータービューというなかなかオツなゲームだ。当たり判定が影の部分にあるとか，斜め移動が普通のゲームと違うとかいろいろあるようだが，独特の世界観からか，なかなかうまくまとまっていると思っている。ボスがもうちょっとやわらかければなあ……と思うことがあるが，世の中にはちゃんとHARDESTでノーミスクリアできる人もいるので，腕に覚えのあるハードゲーマーの人はジョイスティックを磨いて待っていよう。

いま編集部にあるのは開発バージョンなので，X68000版の移植度などは判断できないが，きっとよいものができることだろう。サウンドはまだできていないのでわからないが，グラフィックについてはなかなか期待できそうだ。ドット比をうまく合わせるのが難しいようだが，どうなることやら。　(瀧)

X68000用　5″2HD版　7,800円(税別)
ネクサス インターラクト　☎03(5474)3581

R.C.ロボット集+α vol.4

Oh!Xでも1993年のGAME OF THE YEARでゲームデザイン賞を獲得した「ロボットコンストラクションR.C.」。その後も，パソコン通信や郵送により集められたロボットデータによるバトル大会が続けられている。その大会の結果などを収録したロボット集の発売も，これで4回目となった。

今度のロボット集には，第4回ロボットバトル大会の結果とその記録とR.C.ユーザーが作ったフリーウェアのほかに，「ロボットコンストラクションR.C.」をver.1.11(10秒ルール対応)へバージョンアップするキットが収録される予定である。

発売は12月9日。

X68000用　3.5/5″2HD版　1,800円(税込)
TAKERU　☎052(824)2493

レッスルエンジェルスSPECIAL

「レッスルエンジェルス」シリーズの最新作が登場した。前作「レッスルエンジェルス3」ではシミュレーションゲームだったが，今回は再びシナリオに沿ってゲームが進行するRPG式に戻っている。

プレイヤーはオリジナルのキャラクターを操り，日本女子プロレス界の頂点を目指す。入門したての頃に始まり，海外修行，団体の旗揚げなどを経験しながら次第に成長していく。

試合形式はカードバトルだ。プレイヤーが出したカードの種類や数字，レスラーの能力，出そうとした技などを総合して，どちらの技がかかったかが決まる。このSPECIALではカードをより重くみるようになり，偶然性が強くなった。技の種類が増えたのはもちろんのこと，そのほか新たにパワー技というカードの種類ができて，戦略はいっそう多彩になっている。

試合の結果に応じてポイントがもらえ，5試合ごとに出てくるジムモードで，技の修得や能力の向上に振り分けることができる。関節技のエキスパートや空手系の打撃技を持つキャラなど，プレイヤーの好みに応じて育てることができる。必殺技の登録もでき，さらに必殺技に名前がつけられるなど，プロレスファンにはうれしい気配りがされている。

新たに各キャラクターの勝利グラフィックやセリフがつき，ハマリ度は高い。18禁のパッケージだが，ゲーム内容だけで十分「買い」といえる1本である。 (E.K)

X68000用　5″2HD版　8,800円(税別)
エクシング・エンタテインメント
☎052(824)2493

Ko-Window アプリケーション集6

TAKERUにて発売されているKo-Windowアプリケーション集シリーズは，このほど第6集が追加された。今回のディスク内容は，パズルゲームやサンプルデータ付きの着せかえなどの手軽に遊べるものや，各種の実用的なツールのほか，ザウルスとの通信ツールや手書きメモデータの表示・作成ツールもある。

また，Ko-Windowプログラム入門として，各種マニュアルなども収録されている。

X68000用　3.5/5″2HD版　1,200円(税込)
TAKERU　☎052(824)2493

発売中のソフト

★EXEディスク2　TAKERU　11/20
X68000用　3.5/5″2HD版　200円(税込)
★Ko-Windowアプリケーション集6
TAKERU　11/下
X68000用　3.5/5″2HD版　1,200円(税込)
★シャーペンワープロパック　計測技研　12/上
X68000用　3.5/5″2HD版　6,800円(税別)
★上海 万里の長城　EAV　12/2
X68000用　5″2HD版　8,800円(税別)
★パックランド　電波新聞社　12/9
X68000用　5″2HD版　8,200円(税別)
★R.C.ロボット集＋α vol.4　12/9
TAKERU/エレクトリックシープ
X68000用　3.5/5″2HD版　1,800円(税込)
★レッスルエンジェルスSPECIAL
エクシング・エンタテインメント　12/15
X68000用　5″2HD版　8,800円(税別)
★魔法大作戦　EAV　12/16
X68000用　5″2HD版　9,800円(税別)

新作情報

★VIEW POINT　ネクサス インターラクト　1/8
X68000用　5″2HD版　7,800円(税別)
★X CASE　Béシステム
X68000用　5″2HD版　19,800円(税込)
★ロボスポーツ　イマジニア
X68000用　5″2HD版　価格未定
★Traüm　象スタジオ
X68000用　5″2HD版　価格未定
★鮫！ 鮫！ 鮫！　KANEKO
X68000用　5″2HD版　価格未定
★達人　KANEKO
X68000用　5″2HD版　価格未定
★エアバスター　KANEKO
X68000用　5″2HD版　価格未定
★サバッシュⅡ　ポプコムソフト/グローディア
X68000用　5″2HD版　価格未定
★麻雀悟空・天竺への道　シャノアール
X68000用　5″2HD版　9,800円(税別)
★スタークルーザーⅡ　アルシスソフトウェア
X68000用　5″2HD版　価格未定
★地球防衛MIRACLE FORCE　カスタム
X68000用　5″2HD版　価格未定
★プリンセスメーカー　ニュー
X68000用　5″2HD版　14,800円(税別)
★ディグダグ/ディグダグ2　電波新聞社
X68000用　5″2HD版　価格未定
★フォント＆ロゴ デザインツール
書家万流SX-68K　シャープ
X68000用　3.5/5″2HD版　価格未定

THE SOFTOUCH

●パックランド

我が青春のパックランド

Komura Satoshi
古村　聡

パックマンシリーズのイメージといえば，暗い(？)迷路の中。そんななかで異色を放つ「パックランド」。さあ，迷い込んできた妖精さんを助けてあげましょう。そういえば，これの舞台はパックマン形の島なんです。知ってました？

「パックランド」。忘れてしまった昔を思い出させる懐かしい響きだ。

このゲームがリリースされたのは1984年，いまからちょうど10年前。私は地方都市のすべり止め私立高校に通う高校生だった。当時はどこも校則が非常に厳しくて，私の高校でも，ゲームセンターへの立ち入りは当然のように禁止されていた。何も悪いことなんかしていない，ただゲームが好きで遊んでいるだけなのに，町中を巡回して取り締まりをしている教師は，生徒を捕まえては，学校のコケでぬるぬるの噴水掃除という罰を与えていた。もっとも，当時のゲームセンターはいまよりも薄暗く，タバコの煙が立ち込める場所で，確かに怪しい雰囲気ではあったのだけど。

さて，この「パックランド」はそれまでよくあったゲームと違って，背景もカラー，キャラクターもはっきり目や口，表情が描かれて，まるでテレビアニメーションをそのままゲームにしたような見目麗しいものだった。で，私は一目見たときからこのゲームの虜になり，なんとか昼飯代を浮かせて(B定食320円のところをご飯と生卵だけにして130円ですませたりした)，50円玉を調達しては町なかのデパートの隣の隣にある今川焼屋の２階の，禁断のゲームセンターに向かったものだった。

教師が見回りにきたのを察知して，トイレの窓から外に逃げたのも１度や２度ではない。幸いなことに，私は放課後の噴水掃除という光栄な作業に従事しないですんだのだけど，いまでもあの頃のゲーセンの鉄階段を飛び降りて逃げ出す自分の姿と，抱えて走ったカバンにしみついていた，甘い今川焼の匂いを思い出すのだ。

これが幻の芸術点10点だ

考えてみれば，このパックランドが出た1984～1987年頃のゲームセンターは，変革著しい成長の時期でもあった。「パックランド」のような画面のフルカラー化。「アウトラン」のようにそれだけで成り立つようなBGM。「スーパーリアル麻雀PII」のフル画面アニメ。「スペースハリアー」に代表される大型筐体などが出たのもあの時期だ。

また，PTAなどによってゲームセンターが非行の温床のようにいわれたり，風俗営業法の改正で深夜0時以降の営業が禁止されたり，障害が多い時期でもあった。いまにして思うと，よくあの時期を乗り切ったものである。あの頃はゲーム業界全体にみなぎるパワーがあった。そう，あの頃は何もかもが若かったんだ，何もかもが。

虜にさせるフィーチャー

この「パックランド」は本当に素晴らしいゲームだった。

モンスターたちの攻撃を避けてフェアリーを妖精の国に送り届ける，というストーリーに沿って，画面が横方向にスクロールしていく，当時としては非常に斬新な横スクロール型冒険アクションゲームだった。また，グラフィックの面でも背景，キャラクターがそれぞれに個性的で，いま見ても秀逸なデザインのゲームである。こんなことができたのも，当時アーケードゲームに使われていたマイコンが急激に進化し値段も安くなる，まさに成長期で，搭載できるメモリもどんどん増えていったおかげだろう。

しかし，ゲーセンの便所の窓から脱出させるほどに，このゲームが私を魅了した最大の要因は，そういった力技ではなく匠の技，このゲームでできる日々の新たな発見，フィーチャーだったような気がする。

ある特定の動作をすると，便利なアイテムを手に入れたり，点数が加算されるなどのさまざまなフィーチャーがあって，これらが非常に数多くしかも巧みに隠されているため，プレイするたびに新たな発見ができるようになっているのだ。

たとえば，「ヘルメット」というアイテム

X68000用　5″2HD版　8,200円(税別)
電波新聞社　☎03(3445)6111

比較的簡単に出せるラッキーパック

水柱は使いよう

がある。進んで行く道々には消火栓や木といった，跳び越えなくてはならない固定障害物があるのだが，なかには押すと動いてしまう障害物もときどきある。そして，これを動かし続けているとパックマンにヘルメットが装備されるのである。これは，モンスターたちが飛行機から落としてくる子モンスターたちを受けても平気(普通は死ぬ)になり，しかも時間がたっても解除されない非常に便利なアイテムである。

ほかにも，モンスターをパワーエサでやっつけるときの，その順番によってラッキーパックが取れるとか，障害物だと思っていた水柱に，実は乗って上に行けるとか，面クリアしたときに，ジャンプしていると，その姿勢によって「芸術点」というものが加算されたり(美しくキメると最高7650点のボーナスが入る)，まさにフィーチャーの宝庫だったのである。

ところで，この芸術点にはくやしい話があって，当時ゲーマーの間では，芸術点には最高点7650点よりもさらに出しにくい「10点」という点数があるという話や，うまいタイミングでジャンプすると実はモンスターの2階建てバス(ROUND 5あたりから出てくる)の子モンスターの乗っている2階部分に乗ることができるらしいという噂が流れていたのである。でもって，私はこれらを出すために毎日ゲームセンターに通ったのだが，結局，そのゲーセンから「パックランド」がなくなってしまうまで，それらを出すことはできなかった。「ガセだったのかもしれない」と思っていたのだが，なんと，今回，X68000版でやってみたら……出るではないか！　某スタッフの協力で，写真にもあるように10点を出すことができたのである。なんと奥の深いゲームだろう。10年目にして初めて，10点の看板を見てしまったのである。

ちなみにバスの2階に乗るほうはまだ見ておらず，いまはまだ練習をしているところである。もし，うまい乗り方を知っている人がいたらぜひ教えてほしい。

バスの2階にちゃっかり乗るには？

フェアリーを送り届けると魔法の靴が

ちなみにここでも芸術点が出る

いまに伝える丁寧な作り

で，このゲームをX68000版でプレイして，さまざまなフィーチャーを出現させてみると，感覚的にもアーケード基板と寸分違わぬ感じで遊べることがわかることだろう。X68000版は本当に，画面，動き，アルゴリズム，当たり判定，効果音とどれをとってもアーケード版そっくりにできている。どこが違うのか教えてほしいくらいだ。できれば，画面はHELPキーを使ってディスプレイを15kHzモードにし，それから画面の縦横比を4：3になるように調整してから遊んでほしい。そうすれば本当に基板と見分けがつかない。タイトル画面でF1＋F5を押すと画面に調整用の格子模様が表示されるので，この四角が横4：縦3の横長になるように垂直振幅を調整すればいい。

マシンに関していえば，CPUはオリジナルが6809，こちらが68000と格上であるとはいえ，X68000は決してオリジナルのゲーム基板とまったく同じようにゲームができるように設計されているわけではない。特に，このゲームには19面にパックマンの目の前と，窓になっている一部の部分だけがスクロールして見えて，あとはすべて暗闇になっている面がある。通称「暗闇面」である。X68000はハード的に，マップの上にマスクをすればこのような効果ができるようにはなっていないため，同じことをX68000上で再現するには非常に苦労することになるはずである。ところが，今回の移植では，このような場面でも，10MHz機ではさすがに多少処理落ちするものの，16MHz以上のマシンでは完璧に同じように再現しているのである。電波新聞社は不利を克服してよくここまでやってくれたものだと思う。実はこの「パックランド」はアーケード版以外にも，家庭用ゲームマシンに移植されているのだが，X68000よりもこのゲームに向いているハードと比べても，移植度はX68000版のほうが高い。3つボタンパッドも付属しているので，操作性についても問題はないだろう。

結局のところ，ゲームの面白さを決めるのはハードの強力さではなく，作る人の思い入れと努力，あるいは自分のプログラムへの愛情の問題なのかもしれない。この「パックランド」はそれを証明しているような気がする。あらためて，ゲームについていろいろと考えさせてくれるのである。

これが19面。通称，暗闇面だ

往年の名作と執念の再現力と

「ラリーX」「パックマン」「マッピー」そしてこの「パックランド」に「リブルラブル」と，ユニークなデザインで，次から次へとゲームをヒットさせていたのだから，当時のナムコはすごかったのだなぁ，と思ってしまう。これを考えたら，いまのゲーム業界は「ストII」＆その類似品しかないと思えてしまうくらいだ。特にこの「パックランド」は画面がカラーで描かれているせいか，遊んでみてもいまだに古い感じがしない。

そして，その名作をここまできっちり再現した電波新聞社はすごい。画面の縦横比を調節して遊べば私にはまったくオリジナルと見分けがつかない。まさしく執念の再現だ。
両社に拍手を。

総合評価	0　5　10
BGM	★★★★★★★
グラフィック	★★★★★★★★
操作性	★★★★★★★
処理速度	★★★★★★★★
再現性	★★★★★★★★★

3つのルートの中国旅行

Takahashi Tetsushi
高橋 哲史

おなじみのパズルゲーム「上海」の最新作がいよいよ発売になりました。「万里の長城」は，山海関から出発して中国大陸を進んでいきます。山あり谷あり，パズルを解いて美しい風景を堪能してくださいね。

なにしろ上海なのです。その昔X1版で狂おしいほどに私の頭を悩ませ，ねたましいほどに時間を吸い取ってくれたあのパズルゲーム「上海」がグレートウォール・北京・青島のニューゲームモードをひっさげてX68000に帰ってきてしまったのです。あぁ，あぁ……，ただ牌を取ってゆくだけの単純なゲームなのに……わかっているの……わかっているのに……やめられない，とまらないぃー！　またしても貴重な時間がどんどん吸い取られていきそうなあぶない予感。でも君と一緒ならそれでも……（泥沼)。あぁ，誰か助けてー。

さぁ，嘉峪関目指してレッツゴー！

グレートウォール・北京・青島とは？

ここで上海をご存じない方のために簡単にルールをご説明いたしましょう。「画面上に積まれた牌のなかから同じ図柄同士のものをペアで取り除いていく」。基本的なルールはこれだけです。ただし，取れる牌は「各段の端にあり，なおかつ，上に牌が載っていないものに限る」という条件つきです。こうして牌を順々に取り除いていって積まれた牌をすべて取れればゲームクリアです。文章で書くと単純そうですが，これがやってみるとなかなか奥が深いんですね。油断してるとすぐに手詰まり（どの牌も取れない状態）になってしまいますから。

さてこうやってオーソドックスに牌を取り除いていくいわゆる通常の上海は「クラシック上海」という名称で収録されています。「万里の長城」ではこのクラシック上海のほかにグレートウォール・北京・青島といった3つのゲームモードが選べるようになっていて，それが特徴になっています。それぞれを具体的に説明いたしましょう。

まずグレートウォール。これは上海にテトリスやぷよぷよなどの落ちゲーの要素を加えたモードです。画面上に積まれた牌を取り除くと，その上に載っていた牌は重力に従って落下していきます（落下したあと右や左に動いたりもします)。つまり，牌を取り除くたびに山の様相が変わっていくわけです。うっかりしてると「あれ，この牌はさっきまで取れるようになってたはずなのに？」という事態が続出してパニックしてしまいます。これはなかなか面白いゲームデザインといえるでしょう。ちなみにいくら落ちゲーといってもおじゃまぷよなんか出てきませんので念のため。

続いて北京。これも一風変わった趣向になっています。最初牌はぎっちり長方形に配列されています。「それじゃ真ん中のほうの牌はひとつも取れないじゃないか！」ということになりそうですが，ところがどっこいこの北京では「牌を列ごとスライドさせて同じ図柄の牌を隣接させる」ことによって牌を取り除いていくわけです。これはかなり頭を使わないとどの牌とどの牌が取れるのかわかりません。最初のほうは自分の思惑と違う牌ばかりどんどん消えたりしてあせってしまいます。クラシック上海と同様のルールでふたつの牌をクリックして消すこともできるのですが，スライド取りに比べて極端にスコアが低いため（1/30くらいの得点)，ついつい意地でスライドしてしまうんですが……牌が全然取れないよーっ，しくしくうるうる。これは修行あるのみでしょう。

そして最後は青島。これは万里の長城唯一の対戦上海です。ふたつに分けられた山の底にひそむ黄金牌を先に掘り当てたほうが勝ちという単純なゲームです。相手の山からも勝手に牌を頂戴できるので，相手の邪魔もしほうだいという親切設計になっています（笑)。山の配牌の良し悪しで勝敗が決まってしまいそうな気がしますが，そこ

X68000用　5"2HD版　8,800円(税別)
EAV　☎03(5410)3100

見事クリアすれば，美しい風景が見られるのだが

気を抜くと，「手詰まり」の非情な文字

メニュー式のHELP画面

グレートウォール。牌がどんどん落ちてくる

スライドさせて牌を消す。ちなみにこれは星座牌

はそれ，ちゃんと平等になるように工夫がなされています（対戦を始める前にじっくりと山を観察していただけば納得できると思います）。余談ながら，青島というとすぐビールが飲みたくなってしまう私はやっぱり酒呑みなのでしょうか？

さらに楽しめるアーケードモード

さて万里の長城ではこの4つのゲームモードをさらにオリジナルモードとアーケードモードの2つのモードで遊べるようになっています。

オリジナルモードではステージクリアするごとにほかのゲームを選べたり，難易度レベルの調節ができたり，無制限のヘルプを使用できたりするので，各ゲームの練習用といった意味あいが濃いでしょうか？

しかしアーケードモードに設定すると，各ゲームごとに全30面をクリアしながら先に進むいわゆる面クリ方式になってゲームが進みます（HELPの使用も3回まで）。面をクリアするごとに中国の名所の風景が浮かび上がってきたりして，なかなか心がなごみます（その筋なグラフィックが表示されるような軟派なゲームではないので悪しからず）。また3面ごとにボーナスゲームが入っていて，そこで稼ぐとHELPやBACKの数を増やせるのもゲーム全体にアクセントをつけているといえるでしょう。全30面をクリアするのはなかなか骨ですが，ゴールに待ち受ける認定証を手にするためにはそのくらいの苦労は必要です。頑張りましょう！

X68000版ならでは，か？

ゲームをやりながら，やけにそつなくX68000を使いこなしてるなー，と思ったら，開発はSUCCESSなんですね。それなら「こなれたX68000の使い方」も納得というものです。ということで，ユーザーインタフェイスは極めて良好です。ジョイスティックなしでもマウス対キーボードで対戦できるし（ほんとは本体接続マウス対キーボード接続マウスで対戦ができるようになっていればもっとすごかったのになーってのはちょっと欲張りでしょうか？）。気分を盛り上げるちょっとした演出として牌が消える際のアニメーション処理や効果音もばっちりですしね。このへんはさすがでしょう。音楽もいかにも中国って感じで，あまりにまんま過ぎるのでちょっと笑ってしまうくらいです（でも個人的にBGM TYPE Cがお気にいり）。

ただちょっと残念だったのが，対戦が青島でしかできないこと。クラシック上海やグレートウォール，北京でも対戦ができればかなり面白いと思うんですけどねー。各プレイヤーが取った牌の数を表示するだけでもいいんですけど……うーん，惜しいなー。しかし最初はクラシック上海の2人協力モードを対戦と勘違いして編集の（ふ）さんと早取りで燃えてしまったりもしましたけど（すべての牌を取り終わってから「……いまのって対戦じゃなかったの？」とか気づく始末）。やっぱり昨今のゲーマーを満足させるには対戦モードの充実は必須らしいですよ。

しかし上海の恐ろしいところは，ちょっとした気分転換のつもりで立ち上げたのに，いつのまにか小1時間くらいは平気で過ぎてしまっている……という麻薬的なハマり加減でしょう。その意味では，この「万里の長城」は4つのゲームモードに加えて難易度設定も5段階あるし，なかなか飽きのこない作りになっています。こたつに入ってみかんを食べながら上海……あなたもいかがですか？

対戦上海青島。輝く黄金牌を手にするのはどっち!?

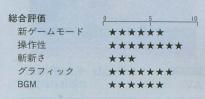
ボーナスステージは神経衰弱だ

パズルゲームの系譜

テトリスにしろコラムスにしろ，ある程度人気の出たパズルゲームには必ずといっていいほど亜種が存在しますよね。テトリスでいえば，「フラッシュポイント」とか「ハットリス」とかとか……。

この「万里の長城」も上海の亜種という点では毛色が一緒ですが，テトリスもどきなんかに比べるとずっと上海のいいところを残しつつ発展させたって感じがして好感が持てます（正直いって，最近テトリスなんちゃらとかいって発売してるゲームは，何がテトリスなんだかわかんないものが多くて閉口します）。さすがにこれ以上バリエーションを増やすと原形をとどめなくなってしまいそうな気がしますが，「万里の長城」はそのへんでうまいバランスを保っているといえるでしょう。

そのバランス感覚に花丸！

総合評価	0　5　10
新ゲームモード	★★★★★★
操作性	★★★★★★★★
斬新さ	★★★
グラフィック	★★★★★★★
BGM	★★★★★★

届け，星の彼方に

Yaegaki Nachi
八重垣 那智

先月号でゲーム内容をご紹介した「魔法大作戦」ですが，残念ながら今月の記事にも製品が間に合わず，評価版でのレビューとなりました。対応機種や推奨メモリなどの動作環境はちょっと厳しいようですが……。

最近ゲームセンターでどんなシューティングゲームを遊んでいるかというと，これが実はいまだに「雷電DX」(ちなみにIIではないので注意)だったりする。「ガンバード」や「ダライアス外伝」は，緻密さが感じられないので，どうものめりこむまでにはいかないというのが正直な感想だ。そこそこ遊べるのはいいけれど，あまりにも覚えてナンボのゲームはいまさらやらなくても，過去にもうたっぷり遊んできたから新鮮味がないという点も指摘しておこう。見てくれだけが進歩しても中身の手応えが感じられなくては，魅力も半減といったトコロだろうか。しかし，だからといって，「雷電DX」が素晴らしいゲームというつもりもなく，これも結局は初心者泣かせの同類ゲームだったりする。このあたりは，五十歩百歩なのだが，紙一重のプレイ感覚の違いで「雷電DX」は何度も遊びたくなってしまうようだ。

ただ共通して気になるのは，商品としての金銭的な足枷が大きいのはわかるが，それならばいちばんお金を使ってくれる人を大事にして作るべきである，ということだ。どれだけ間違っても100円で不必要なほど延々と遊べる仕掛けを作ることに汲々とするのは，商品のクリエイターの意識としては下劣だと断じざるをえない。パソコンソフトについても，そういった原則は少しも変わることはないハズで，あたりまえ過ぎてわざわざ書くのも恥ずかしいことだが，ソフトというものは，それを求めてお金を使う人，つまり(すべての)ユーザーのために作られるべきものなのである。

X68000用　5"2HD版　9,800円(税別)
EAV　☎03(5410)3100

Xのために鐘は鳴る

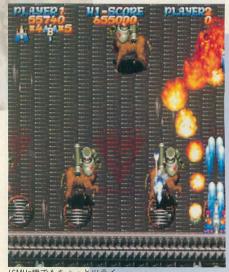
先月号に引き続いての，EAVの「魔法大作戦」のレビューであるが，このゲームの内容を復習すると，一風変わった雰囲気を持つ正統派のシューティングゲームといった感じになるだろう。操作はレバーの移動に，2つのボタンによるショットとボムといったオーソドックススタイルで，4人のキャラクターのなかから選んだプレイヤーで3種類のサブショットを駆使して進んでいくという展開になっている。オリジナルが1993年に発売された業務用ビデオゲームであることを押さえれば，このゲームについての基礎知識としては，とりあえず十分だろう。

そこで今回のX68000版の登場となるのだが，今回の移植は残念ながら全機種対応の移植ではないことを最初に書いておかねばならない。推奨機種ではなく，対応機種がX68000XVI以降となっているので，要16MHzクロックだと思って差し支えない。推奨環境となると，X68030のメモリ4Mバイト以上でハードディスクにインストール済みの状態とあるから，これはかなり贅沢な動作環境を要求するゲームであるといえるだろう。

もちろんX68000シリーズには輝かしい互換性が備わっているので，10MHzクロックの機種でも動作はするのだが，処理が複雑なためなのか，スローモーション状態になってしまい，キャラクターが消えるなどして，まともにはプレイできない。グラフィックがオリジナルからストレートにコンバートしてあるせいか，動きが無残になった画面を見ると，余計に惨めたらしく感じられてしまうのだが，これはあくまでも仕方のないこととして諦めるべきだろう。

また，同時に参加できるプレイヤーも，オリジナルの2名に対し，同時プレイが廃止され，ひとりプレイに限られている。オリジナルのアーケード版では2Pプレイ時の面クリアデモにおいて，登場しているキャラクター同士が掛け合いをする「ボケ・ツッコミシステム」なるものが搭載(!?)されていたのだが，結果的にこういった部分の演出要素が丸ごとカットされているというのは，ちょっと残念なところである。カットの理由は，同時プレイによる処理の負担の増大にあると考えられるので，スローモーションやキャラクター切れを覚悟のうえで，強引に遊べるようになっていてもよさそうなものだが，これも難しいところではあるようだ。

いまそこにあるギャップ

こうした環境に対する制限から，やや厳しいマイナスイメージの印象を受けてしまう，X68000版の「魔法大作戦」であるが，表面的な部分だけでなく内容に関する制限

16MHz機でもちょっとツライ

も少なからず存在している。続けて，それらを含めた移植の質的側面を検証してみることにしよう。

まず，気になるのは画面サイズである。タイトル画面が無残に切れてしまっているので，簡単に気がつくと思うのだが，この「魔法大作戦」の縦方向のドット数は256ドットを越えているのは間違いない。はみ出た部分はどうするかというと，今回の移植では画面の上下を切り落としたような処理がなされている。このため画面が狭く，比較的縦方向に高速に動くモノ(特にパワーアップアイテム関係)の多いこのゲームでは，画面に存在している時間がその分だけ短くなることで，かなり大きな影響がゲームに現れている。

アイテムは取りにくくなり，敵を攻撃できるチャンスが減ってしまう，と書くと簡単だが，ゲームの難度が密かにアップしていることは間違いない。しかも画面の移動範囲も一緒に小さくなっているわけだから，ボス戦では必然的に敵に近づいて戦うことになる。これらが思った以上に，オリジナルのプレイ感覚との間にギャップを生じさせてしまっている。こういった部分は敵の耐久力などの調整でいくらか緩和されるような気もするのだが，そういう対策は立てられていない。オリジナルのデータを尊重する姿勢もよいのだろうが，結局プレイ感覚で違和感を与えてしまっていては，本末転倒だといえるだろう。

次に挙げるのは技術的な部分で，表面的にもハッキリしているのだが，全体的にキャラクターの表示の負担が重いという点である。今回主にX68000XVIを用いてプレイしたが，頻繁なキャラクターの点滅の発生に加えて，自機すら消えてしまうというような重症のキャラ消えすら決して珍しくなかった。ボスキャラクターと対戦中に大量の弾をバラまかれて自機が消えてしまっては，ゲームどころの騒ぎではないだろう(もちろん表示が消えているだけで判定はある)。これが10MHzの非対応機で強引にプレイした結果だというのならば我慢もするが，まがりなりにも対応機種と明記されているのだから，こればっかりはいただけない。なんらかの対策が欲しいと切実に願うところである。

ほかにも，ボスが出現する直前で自機が勝手に動いたりする気色の悪い動作も，いくつか見られた。スクロールの不規則なガタつきのように明らかに不具合を感じてしまうようなものは，実害がなくても決してよい印象にはなり得ないものだ。こういった部分にもツメの甘さが目立つのは，やや残念なところではある。

解体(バラ)しちゃうぞ，とはいわない？

奥様ウットリ極太レーザー観覧席，重要!!

ゲーム難度も高くなっているようだ

華麗なる移植

X68000シリーズには，いくつか縦画面シューティングが移植されているが，過去の作品と比較して，「魔法大作戦」というゲームがそれほど移植が困難だという印象は受けなかった。しかし，実際には疑問符がついてしまったことには，かなり残念な気持ちがしている。特に「コットン」のクオリティが記憶に残っていて，ついつい過度の期待をしてしまった点については否めないが，プレイ感覚の部分が比較的軽視されている移植の観点にいちばんの原因があるのだろうと思われる。

冒頭にも書いたが，商品としてのソフトは，あくまでもユーザー本位のものでなくてはならない。移植型のゲームソフトにユーザー(プレイヤー)が期待するのは，そのプレイ感覚であり，「本物と同じように」遊べるという事実である。そういった意味では画面がドット単位で一致している必要はどこにもないし，BGMがアレンジ版になっていても，プレイ感覚さえ大事にしてあればユーザーが感じるギャップは最小限で済むハズである。今回の「魔法大作戦」ではまさにこういった，感覚的なギャップを埋めるための工夫のない分，それがかけ離れたモノになってしまったのではないのだろうか。移植というのは何を移植するためのものなのか，改めて考えさせられたソフトであった。

いらっしゃいマホー(死語)

個人的にはこの手のゲームは，攻撃力の強いキャラクターか誘導攻撃の手軽さで遊んでしまうので，使うのはチッタかボーンナム。しかし画面が狭いおかげでイマイチ敵に押されるのでボーンナムでもちょっと敵が硬く感じてしまいました。あと，いちばん下まで降りてこないハズの敵に画面下で踏みつぶされたのが何度か。

「究極タイガー」のときから進歩してないようだけど，確固たる自分が完成しているっていえばカッコいいな，そうしよう。うんうん。

総合評価	0 — 5 — 10
ゲーム性	★★★★★★
グラフィック	★★★★★★★★
技術	★★★★
サウンド	★★★★★★
爽快感	★★★★★★
チッタ(笑)	★★★★★★★★★

特別編8

熱血パンツ男の来襲 Joe Higashi

Nishikawa Zenji 西川 善司

ジョー東は餓狼伝説SPECIALでは実にスタンダードな実力を持っている。上手い人が使えば強いし，下手な人が使えば弱い。君もジョー東をマスターして熱血男になろう。

お手軽連続技

連続技と呼ばれるものはたいてい高度なテクを要するものなのだが，これから紹介するジョー東の連続技は非常にお手軽に決めることができる。密着して小パンチを連打するだけだ。小パンチが数発入ったのち，爆裂拳が始まり，これがほとんどすべて入る。途中で逃げられてしまうこともあるが，そのまま削りモードに突入するだけで，いずれにせよ相手の体力は奪える。

爆裂拳は跳び込み攻撃からつないでもいいし，しゃがみ攻撃からつないでもいい。とにかく何か技が決まったら爆裂拳につなぐといい。小パンチ連打から立ち大パンチというパターンもある。いずれにせよ技術はいらない。お手軽だ。

無敵対空技タイガーキック

出だしは完全に無敵。飛び道具も抜けられる。転ばされたときに起き上がりを攻めてこられたら，もうタイヤキィッ（タイガーキックのことね）しかない。逆に，敵の体力が風前の灯のときも，相手が転んだならば相手にジョー東の体を重ねて大タイヤキィッをお見舞いしたれ。

無敵の対空技，削り技として君臨しているタイガーキックだが，敵のスピーディな跳び込み攻撃にとっさに反応できないときもある。そんなときは立ち大キック（回し蹴り）も役に立つ。もちろんタイヤキィッには及ばないが，立派な対空技になるぞ。やや早めに出すのがコツ。

ホモパンツのほうが弱いらしい（嘘）

空中戦はお任せジャンプキック

空中で膝を前方へちょこっと突き出した，この地味な技ジャンプ小キックが強い。敵と空中で交差した場合，これを出しておけばまあ大丈夫。地上を滑る十兵衛のスライディングにも勝つことができるぞ。

このジャンプ小キックで跳び込み攻撃が成功したならば，そのまましゃがみ小足払いにつなげて，さらに小パンチを連打すれば爆裂拳までつながる。最後の爆裂拳は単なる削りになってしまう場合がほとんどだが，敵は以後ジョーと接触するのは決して気持ちのいいこととは思わなくなるだろう。

空中キックでもうひとつ強いのがジャンプ大キック。足がニョキッと伸びていくのが特徴的。飛び道具を跳び越えて相手を空中から襲うときの必須技だ。

小を選ぶか大を選ぶかは，敵の攻撃パターンや性格に応じて臨機応変に。

たいやきぃ～ってとこ

使える必殺技と使えない必殺技

タイヤキィッはなかなか使えるが，大はガードされたときの隙が異常に大きい。出すときは覚悟とそれなりの「確信」が必要。

同じキック系必殺技にスラッシュキックがあるが，足元が意外に弱くあまり使えない。奇襲や間合い詰めにはいいかもしれないが，ガードされると相手にめり込んでしまい，これまた隙が大きい。

ハリケーンアッパーは，発射コマンドがほかのキャラの飛び道具よりやや難しめなのが難だが，撃ったあとの隙は小さいので多用もOK。ハリケーンで飛ばしてタイヤキィッで落とすというパターンも使える。

超必殺技のスクリューアッパーは，出すのがやや難しいうえ，出るまでに時間がかかり，相手のほうへ少し進むと消えてしまい，まったく使いものにならない。ただ，撃つときのモーションがハリケーンに似ているのでこれを利用する手もある。ハリケーンを連発し，相手にこれを印象づけて，突然スクリューを出すのだ。たまーに，ただのハリケーンかと思ってジャンプ攻撃してきた敵に大命中することがある。もちろんこれは「超」必殺技なので，起き上がりに相手に体を重ねればゴスゴス削れるぞ。

ムエタイチャンピオン，攻撃開始

ち，ちんはどこ？

タイヤキッができないときはスライディングで

特別編9

完成されたリョウ・サカザキを見よ!

Ryo sakazaki

Taki Yasushi 瀧 康史

シリーズ最強のリョウ・サカザキ!

ストーリーモードをノーミス(一度も負けないこと)でクリアするとクラウザーのあとに出現する最強の男リョウ・サカザキ。いわば，隠しキャラ的存在だ。

さてこのリョウ・サカザキ。ストーリーモードでもvsコンピュータモードでもプレイヤーは使用できない。使えるのは対戦モードだけだ。したがって，極めても対戦相手がいないとプレイできない。

はっきりいってリョウ・サカザキは強いキャラだ。ただし，プレイヤーが使うとコンピュータのような圧倒的な強さはないので，いい加減に技を出しまくっていては負けてしまう。餓狼伝説SPECIALでの最強キャラは，キム・カッファンだろうけど，その次くらいに位置するだろうか。

大技! 大技! 大技!

というわけでリョウの長所と短所を挙げていこう。キャラを極めるには，そのキャラの特性をよく知ることが大切だ。

まず，リョウの技はすべてが強く，使えそうなものばかりだが，すべてにおいて大技だ。そして，実はどれも連続技にはなりにくい。これが弱点。遠めの大キックと大パンチ以外は全部の通常技がキャンセルできるので，使いにくいわけではないが，必殺技はすべて出だしが遅いか，あるいは出したあとに無防備になってしまうので，いまいち使えない。

では，必殺技をひとつずつ紹介しよう。

まず虎煌拳。この飛び道具はかなり速く，

虎煌拳はひたすら速い

硬直も少なくて使える技なので，敵を隅に追い詰めて虎煌拳を連射すれば，対処方法を知らない相手ならばそれで参ってしまうほど。しかし，出だしが若干遅いので連続技にはなりにくい。

次にビルトアッパー。いわゆる昇龍拳で，出だしの瞬間だけ無敵だが，無敵時間が短いので飛び道具をすり抜けるのはかなり難しい。地上で当たれば3発当てることはできるが，敵は当たっても倒れることはない。最も連続技になりやすい技だけど，これでフィニッシュか，ピヨにさせない限り，必ず反撃を食らってしまう。キャンセルさせる技は立ち小キックなどがお勧め。キャラによってはしゃがみ大パンチから入るのだが，結局のところ，あとのことを考えるとあんまり使えない。これで，ビルトアッパーを食らった敵を転ばせることができたなら，最強キャラだったのにな。

飛燕疾風脚は，タメなしでできる強力な技。不意に出されると怖い技なんだけど，1段目と2段目には隙があって，1段目を食らえば，ギースなどは当て身投げができ

でっかい覇王翔吼拳。当たると痛い

てしまう。出だしも速いので，しゃがみ大アッパー-飛燕疾風脚は2段連続技にはなるけれど，3段目で反撃を食らってしまうので，ダメ。虎煌拳で隅に追い詰めたときに，敵がジャンプしてかわしたら飛燕疾風脚を出す。この程度しか使えないかなぁ……。

そして覇王翔吼拳。出だしに構えがあるので，連続技には絶対使えない。ただし，ジャンプでかわすのは難しいほど大きな飛び道具なので，1ラインステージなどでうまく使えばいい。虎煌拳と織り交ぜて使うと，相手は引っかかってくれるかも。

最後に龍虎乱舞。これは強力な元祖乱舞技。最初に覇王翔吼拳と同じ構えをとり，(したがって，一度覇王翔吼拳で誤魔化して龍虎乱舞という方法もある)ダッシュで突進して乱舞を決めるというもの。判定は投げと同じようなので，相手が防御していても決まる。なんと，空中にいても頭がカスりさえすれば決まってしまうのだ。

ただし，ダッシュ中は無防備なので，小パンチなどを出せば返されてしまう(速いから難しいけど)。だから，攻撃判定を出しながら迫ってくるキムの鳳凰脚とお互いに出し合ったら負けてしまう。

うまい決め方は，大キックなどからのキャンセル。敵に見切られると小パンチで返されるので，ときどき覇王翔吼拳を出す。それでなくても，相手の恐怖感はただごとではないんだからね。

とにかくリョウを使うなら，うまく大技を活かすこと。フェイントはどのキャラでも必要だが，サカザキはとにかくフェイントがうまく効いたときが快感。

やはり，派手に決めたきゃリョウ・サカザキってとこかな。

龍虎乱舞vs鳳凰脚の決着は……?

実はキムが勝つんだな

特別編1

いつも笑顔のマシンガンダンサー

Nishikawa Zenji 西川 善司

Dee jay

スーパーストIIの新キャラ，ディージェイは，必殺技よりも通常技の使い方の「上手」「下手」が「強い」「弱い」に直結する。というわけで，ここでは彼の持つ通常技の威力について分析していこう。

強力理不尽空中小パンチ

空中で敵と交差した場合，絶対に役に立つのが空中小パンチだ。威力はないが攻撃判定がやたらに強く，しかも自分のやられ判定は異様に小さい。もう「空中はコレで決まり！」って夜7時あたりにテレビコマーシャルしたいくらいだ。

これを出しながら相手に跳び込むのもいい。ガイルのサマーソルトキックや，リュウ，ケンの昇龍拳には負けるが，一般的な通常対空技にはたいてい勝てる。最悪でも相撃ちだ。ブランカの必殺技バーチカルローリングにも勝てたりする。

相手の跳び込み攻撃も，早めに反応して垂直ジャンプ→ジャンプ小パンチができればほとんど防げちゃう。

とにかく理不尽なまでの強さは一見の価値ありだ。

強力不条理しゃがみ大/中パンチ

空中最強のジャンプ小パンチを持ったディージェイは，さらにずるいことに地上最強のしゃがみ中パンチを持っている。

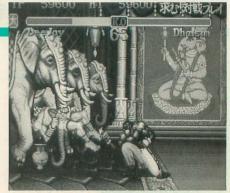
この中パンチは見た目より前に当たり判定があるようだ。具体的にはV字に突き出した腕の先方。敵が歩いてきたときにパシっと決めてやるのがスタンダードな使い方だ。ザンギエフなどは接近戦してなんぼのキャラクターだが，こういったキャラは，やたら間合いを詰めてくる。この技はコイツらの接近を阻むにはもってこいなのだ。

さらにこの技，やられ判定がこれまた小さい。そのためか敵の足払いを一方的に軒並み潰すことができる。ガイルの足払いはもちろん，ブランカのローリングアタック，バルログのゴロゴロなど，必殺技も簡単に撃退できてしまう。

しゃがみ大パンチもほぼ同様の威力を持っている。技のアクションが大きいのが欠点だが，ブランカやバルログのローリングもちゃんと潰せる。足払いも潰せる。とっさにはしゃがみ中パンチ，余裕があるならば攻撃力の大きいこちらを使えばいい。しゃがみ大パンチは座っている体勢から一瞬起き上がるモーションが入るのでフェイントにも使える。相手が「む，エアスラッシャーか」と思ってくれたらしめたもの。

そのほか使える通常技は

●**立ち大キック**

必殺技マシンガンアッパーは確かに威力のある対空技だが，敵の奇襲にとっさに対応するのは結構難しい。また，仮に敵にヒットしても，百裂系の技のため攻撃力の1発1発が小さい。これよりも対空攻撃に適しているのがこの立ち大キックだ。「大」攻撃のため技が完全に出るまで時間がかかる。よってやや早めに反応するのがコツだ。

あまりにも強いヒジ撃ち

●**しゃがみ中キック(足払い)**

ヒットすると確実に相手を転ばすことができる。近づいていって急に出すとか，自分が端に追い詰められたときに，相撃ち覚悟で出すのもいい。当たれば必ず間合いを広げられるからだ。攻撃判定がしゃがみ姿勢の割にはずいぶんと前に発生するので，対戦などでほかの技を空振りしておいて出したりと，トリッキーな使い方もできる。

●**しゃがみ大キック(スライディング)**

相手を転ばすことができるのは見た目どおりだが，実は，前進→対空のワンセット技としても使える。相手のジャンプからの着地点へこれを重ねると，あら不思議，敵は着地できずにスっ転ぶ。あと，これで地上からの竜巻旋風脚も一方的に潰せることも覚えておこう。

●**ジャンプ大キック**

空中で早めに出せば絶対的に強い。バルログの「ヒョー」も潰せるし，飛び道具を撃った敵の硬直に向かってこれで跳び込むのもいい。密着した状態でヒットすれば，そのままレバー横＋中パンチ連射で投げへとつなげられちゃう極悪技にもなる。

●**垂直ジャンプ小キック**

敵の起き上がりにこれを重ねるのがセオリー。小攻撃なので威力はないが，前述の空中小パンチ同様，反撃の糸口をつかむきっかけにはなる。

また，斜めジャンプ中に，レバー下＋小キックで春麗の踏みつけもどきが出せるが，これも判定が強く，ブランカのバーチカルなども潰せるぞ。

ずるいほど強い小パンチ

これがウワサの大マタ開きだぁ

特別編2

ホークで始めるスパII入門

T. Hawk

Shirai Isao 白井 五三雄

スパIIでいちばん貧乏なのはいったい誰でしょうか？ T. ホーク（以下ホーク）ではないかと僕は思っています。靴を履いているのに騙されてはいけません。彼は故郷を奪われて，すっかりホームレス復讐鬼と化しているのです。彼に対抗できるのはブランカくらいでしょうか？（リュウは本当に貧乏かどうか怪しい）

コマンド系キャラの集大成

ホークの特徴を簡単にいってしまえば，昇龍拳つきザンギ。こいつを使いこなせればほかのコマンド系キャラの基本はたいてい大丈夫なはずであり，飛び道具に頼らないプレイも練習できます。それに，なんといってもユーザーが少ない。対戦においては，リュウやガイルで始めるよりも短期間で友達との差を縮めることができるお勧めのキャラ。どうです。やりたくなってきませんか？

技のコツ

まずはメキシカンタイフーン。ほかのすべての技はこの技を出すためのおまけにすぎません（そのおまけの部分が充実するほど，味わい深いプレイができるのですが，今回は触れません)。個人差はあると思うのですが自分が2P側（右側）に立ってレバーを時計回りに回すのが楽だと思います。それではやってみましょう。 まず，デク（練習台）を1P側に立たせます。そしてホークを技のかかる間合に立たせます。そして次が重要です。レバーを右から時計回りに回して上にきたときにパンチ（初めのうちは小でいい）を押すのです。あれ？ と思った人もいるかもしれませんが，スパIIのレバー1回転技は，4分の3回転でいいのです。上にきたときに押すと思えば，失敗のジャンプも減るでしょう（ザンギも同じ）。これがいわゆる“立ちタイフーン”って技なんですが，これができればそのほかのバリエーションも簡単にできるようになります。ぜひ身につけてください。

インディアンでもモンゴリアンチョップ

次はトマホークバスターです。ホーク式昇龍拳なわけで，出し方のコツもリュウと同じです。ポイントは最初と最後にあります。最初のレバー右（自分が1P側）のときにジャンプしないことと，最後の右下のときにちゃんとしゃがむところです。どうしても出ない人は，まずレバーだけで練習してみましょう。最後の部分をちゃんとしゃがめるようになったら，小パンチを押してみます。一瞬しゃがみのグラフィックが出てから小パンチを押す感じです。がんばってください。

飛び道具や足払いはこのキックで返せ

始めてみよう

どちらも出せるようになったらCPU戦で練習です。跳び込んでくるのを待って，それをトマホークバスターで落とし，のこのこ歩いてくるようならタイフーンで吸い込みます。どうです。おそらくできないでしょう（できた方ごめんなさい)。デクと動く相手は別物なのです。しばらく特訓してみてください。それができれば，いよいよ人間と対戦です。

対戦

安易な跳び込みはしてこないし，タイフーンの間合いで勝負はしてこない。しかも飛び道具で押されて，苦し紛れに跳び込んだら迎撃されてしまった。

と，まあ初陣はこんな感じではないでしょうか？ そこで考えることはいろいろあるのですが，とりあえず注意することは，いかに跳び込ませるか，そしていかに飛び道具を封じるかの2点です。通常技をメインにして，自分なりに研究してみましょう。例を挙げれば，「相手の足払いの当たらない間合をキープして飛び道具を撃とうとするならば立ち大キックで相打ちを狙う」などです。

なんでもそうでしょうが，敗因とその反省が重要です。いきなり闘って勝てる人なんてごく少数です。練習して強敵（とも）を超えましょう！

ちゅーちゅー吸って，ぐるぐる回せ！

頭の硬さなら負けないぜ！

特別編3

血まみれ人生道場

Zangief

Sudou Yoshimasa 須藤 芳政

スーパーストIIになってから，各キャラの声が独立した音声になったため「うわあぁ」「ふうっ」「えい！」「ワッハッハッハ」などのキャラ間で重複していた声がなくなり，コンペだかマンボだかコンボだかいう連続攻撃のカウント表示は，

「これ俺が新しく発見した連続技だよ～」

「ウソつくなよ！　つながってないだろ！」

「何を！　もう一度いってみろ！」

「ああ，何度でもいってやらあ！」

という「あばれはっちゃく風論争」に終止符を打ちました。不満があるとすれば地図のオーストラリア大陸にコアラちゃんが描かれていないことぐらいですね。

なにはともあれ，スパIIでザンギエフ(以下ザンギ)は本当の意味でザンギになりました。「お～りゃ！」「ハッハハ……(イヤラシイおやじ笑い)」などの声に酔い，血まみれになった彼の顔面に心が燃える！

彼のボディプレスで窒息死したいと願う男は数知れず。さあ，キミも「今世紀いちばん吸い込まれたい男」を使いこなせ！

ぎえっ！

さて，ザンギを愛する貴方がやらねばならないことはバンジージャンプでもバンバンボールでも棒棒鳥(バンバンジー)でもありません。立ち小キックスクリューの練習です。

「すいませーん！　それとバンバンのボケと何の関係があるんですか？」なんて質問は説明に困るのでしないように。やり方は相手の近くで立ち小キックを食らわし(密着キックはダメです)，その間にスティックを1回転させてパンチボタンを押すという，ザンギ使いなら誰でも使っている一見簡単そうなこの作業。実はタイミングを体が覚えるまで辛い鍛錬の日々が続いたため，私は，初めてこれが出せたときは感動のあまり，

「はひ～！」

などとわけのわからないことを叫んだ記憶があります。スティックは上側から先に回し始めたほうが間違ってジャンプしてしまったりせずに上手くいくかもしれませんね。

ダブルラリアットの出初めは無敵

立ち小キックスクリューのいやらしい実用例には，次のようなものがあります。

・竜巻旋風脚をしゃがみ小チョップで落としたあと

・春麗などが跳び込んできたところを，こちらもジャンプして，チョップなどで撃墜したあと

・顔面つかみなどのマッサージ技で相手を離したあと

ときには，盆踊りだってします

コラッ！ どこに手置いてるんだあ？

今回は空中投げもOK！

ぎえっ！ ぎえっ！

スクリューのキック版とでもいうべき新技は，忘れた頃に使うと効果的。ガイルが「ソニッ……」までいったときにつかんで，タッチダウンしてしまうのには笑えました。ただ，新しく備わった空中投げなんですが，私はいままで「偶然投げてしまった」以外に決めたことがないんですけど……。

せっかく連続技カウントがついたのだから連続技を出してみようということで，おサルさんでもできる連続技。めくりボディプレス＋しゃがみ小チョップ×3＋しゃがみ大キックで合計5ヒットコンボ！　スパIIでは簡単に出せます。ちなみに私はこれをゲームセンターで連続で決めるたびに次の瞬間ハメ殺された記憶が2回ほどあるので，あまり使ってはいけないようです。しゃがみ小チョップ＋立ち小キック＋立ち中チョップが連続でつながることは前から知られていましたね。

私はスパIIになってから必殺技以外にも通常技を使う機会が増えました。意外と使えるのが立ち中キック。相手と少し間が開いているときにサッと中キックを出すと相手が何かしようとしたところにヒットしてなんだか嬉しい。特にザンギ同士の対戦は通常技のオンパレードですね。お互いにあまり近づきたくないんですよ，コレ。

いい忘れるところでしたが，これからの季節，使うなら脂の乗ったターボザンギの色を選択しましょう。

サンダーホークとかいうパクりスクリューを使うホタテ男のそっくりさんは結構怖いなぁ。でも，負けるもんか！

特集

割り切って使う CD-ROM

CONTENTS

CD-ROM。
大量のデータ。
そしてリアルな音声&BGM。
最近巷ではCD-ROMメディアによるソフト供給も一般化してきました。雑誌類の付録にされていることも多いのは皆さんすでにご存じのとおりでしょう。
こういった状況はX68000を除いたあらゆるところに展開しています。
ドライブはずいぶん低価格になってきましたし，X68000用のドライバも一応市販されてはいます。しかし，アプリケーションがありません。すでに市販されている「フリーソフトウェアセレクション1&2」を見ても，容量が大きいこと以外に媒体がCD-ROMである必然性はまったくありません。
X68000におけるCD-ROMメディアは単に大きな「容器」としての意味あいしかありません。
よってユーザーにとって当面の課題は，汎用，または他機種用のデータをいかにうまく使うか，という点に集約されます。
プログラム自体は機種やOSに従属する部分が大きいので，流用することはほとんど不可能ですが（ソースがあれば話は違うが），グラフィックデータ，サウンドデータなどは機種によらず利用できるものです。
こういった状況はデバイスがちゃんとサポートされてくればまったく変わってきます。しかし3.5インチFDでさえまともにサポートできていない状況ですから，メーカーにそれを望むのは酷というものでしょう。
X68000にとって，メディアとしてのCD-ROMはこういった構造的な問題を抱えつつ，少しずつ着実に普及しつつあります。
「そこにCD-ROMがあるから」
切り開かれるべき未踏の大地は我々の眼前に広がっているのです。

あなたにCD-ROMは必要か?

CD-ROM導入の心得

CD-ROM

Nakano Shuichi 中野 修一

時代はCD-ROM……などとはいいません
万能視されがちですが，使い方次第なのはほかの周辺機器と同じです
導入の際に必要なる基礎事項をまとめてみましょう

12月号の瀧氏の記事にOh!X関係者のCD-ROM所有率は90%であるような表が載っていましたが，これは（控えめにいっても）統計の魔術というやつです。「X68000用の」と書いてないあたりが非常に怪しいので，勘のいい方はおわかりでしょう。私もSCSI対応でないのを（ゲーム機ともいう）4つ持ってはいますが……。

はっきりいってX68000ユーザーの，CD-ROMドライブの普及率は低いようですし，対応のCD-ROMソフトというのも非常に限られているのが現状です。

ストリーマとレーザープリンタを比べたり，スキャナと3.5インチFDDを比べる人は滅多にいないのですが，周辺機器として「ハードディスク＋1」という選択をした場合，CD-ROMはよく光磁気ディスクと比較されます。もちろん，これらは用途的にはまったく異なったものです。まあ，あえてどちらかを導入するというのなら光磁気ディスクが先かもしれません。

結局はパソコンの使い方次第ですが，X68000の場合は他機種とはちょっと違った事情があります。それはX68000で扱うCD-ROMは基本的に他機種用のものとなることです。そのなかから一部のデータのみが利用可能となります。

通常は「初心者でも簡単～」とかいうセリフと対になってCD-ROMメディアが普及しているのですが，X68000上でX68000用でないデータを活用するためにはそれなりの知識も必要とされます。それに対して光磁気ディスクはセットアップさえしてしまえばハードディスクの延長として誰でも使いこなすことができます。それゆえ，CD-ROMというのは非常にマニアックな周辺機器となっています（使いこなすには）。

ですからCD-ROMと光磁気ディスクでどちらかということになれば（こういう議論自体がおかしいのですが），「光磁気ディスク」という結論になってしまっています。

要は簡単に活用できるCD-ROMがどの程度あるかなのですが，最近のCD-ROMドライブの低価格化には，あまり使い道はないことは承知のうえで導入するのも悪くないかな，と思わせるくらいの勢いがあります。定価で1万円台の製品から，X68000との接続に必要なものをすべて込みで2万円台の製品まで出てきているのですから。

満開製作所のように「YELLOWS再生装置」と割り切ってもかまわないならば，CD-ROMは手軽に導入できる周辺機器として取り上げることもできるようになります。

ドライブ自体は安くなっていますから，ひとつ買ってみるのも一興ですが，ただでさえリスキーな周辺機器です。なるべく無駄金を使わないようにするのは大切なことでしょう。そしてなにより，

「多大な期待はしないこと」

に尽きます。

それでは，覚悟を決めて異機種用データ活用の道に分け入るために，または勢いで買ってしまったものをより有効に使うために，X68000でのCD-ROMの扱い方を見ていきましょう。

CD-ROMの基本

音楽用CDは音楽データが媒体上にデジタルで記録されているわけですが，同じ媒体にコンピュータで扱うようなデジタルデータを記録したものがCD-ROMです。

CD-ROMの特徴というと，よく，

「容量がでかい」

「遅い」

ということが挙げられます。

容量は700Mバイト近くまで入ります。可搬メディアでこれ以上の容量を持つものもないではないのですが，コストと手軽さを考えればこれほど効率のよいメディアもないでしょう。

速度のほうですが，ランダムアクセス性能を示すシーク速度とシーケンシャルアクセス性能を示すデータ転送速度があります。

CD-ROMは容量を稼ぐためトラックの構造がCLV（線速度一定）で渦状（スパイラル）というFDなどに比べると複雑なものになっているので，シークするのに多少時間がかかります。直径が大きめなのもシーク速度には不利な条件です。そもそもが音楽用CDの再生のために規定されたメカニズムですから，それほど高速動作に対応してはいません。10ミリ秒単位で次の曲の頭出しをする必要などはありませんから。

一方，CD-ROMのデータ転送標準速度というのは音楽CDの再生速度（回転数）にあわせて決められていました。

CDのサンプリングレートが44kHzということは1秒間に44000個の信号を送っていることになります。ひとつの信号は16ビットなので，だいたい88Kバイト/秒になることがわかります。これがステレオで入っていますから，トータルで秒間170Kバイトくらいの情報量のオーディオデータを扱っていることになります。

デジタルデータの場合，データ補正の強化などで多少信号が冗長になることもあり，CD-ROM標準速ドライブの転送速度は150Kバイト/秒といわれています。

ただ，これではパソコンの記憶装置として見たときには，あまりにも遅すぎます。最近流行の動画などを行うためには転送速度は速ければ速いほどよいのですから。ということで，現在では倍速ドライブが標準となっています。これは回転数を2倍にしたもので転送速度も300Kバイト/秒となっています。現時点では標準速のCD-ROMドライブはまず見かけません。

こういった趨勢のなかで，わざわざ転送速度を標準速のCD-ROMにあわせているMDデータという規格もありますので，もしかしたら常人には理解できないところに150Kバイト/秒という数字の特別なメリッ

トがあるのかもしれませんが。
ちょっと脱線しますがよく聞かれる「CD並の音質」というのは16ビット44kHzのPCMデータのことを指します。これを実現するためのデータ量は1秒間に88Kバイトです（モノラル時）。2秒程度のサンプリング音が7個もあれば優に1Mバイトを超えてしまいます。カタログスペックはともかく，次世代ゲーム機の音質が「CD並」ではないことは明白ですよね（たいていサウンドRAMは512Kバイトになっている）。

CD-ROMドライブの選び方

CD-ROMのインタフェイスには，
1) SCSIのもの
2) IDEのもの
3) 独自のもの
があり，それぞれのドライブが発売されています。
1)にはMacintosh用をはじめとして，数多くの製品があります。国産機を中心に扱っている店で通常売られているのはSCSI対応の製品です。X68000に接続できるのもこれに限られます。
2)はAT互換機で内蔵用に多く用いられているものです。ベアドライブを買ってつなごう……という人以外は目にすることはないので間違えないとは思いますが，当然X68000には接続できません。
3)はローカルなI/Oに接続するタイプのものです。FM TOWNSが代表例ですが，SOUND BLASTERの一部には独自のインタフェイスでCD-ROMをつなぐものがありますので，それ用のドライブには注意が必要です。ゲーム機などのものも独自規格です。普通にX68000につなぐことはできません（知らずにジャンクのドライブを買ってきたのが約1名……)。
実際の製品を選ぶときには，まず，ドライブの内部ユニットのメーカーを調べます。製品自体を発売しているメーカーではなくて部品のほうが重要です。
現在のところ無条件でおすすめできるのはソニー，東芝の2社だけです。「オーディオデータの読み込みができること」というのが最大の理由ですが，そのほか計測技研のドライバと相性がよい点も見逃せません。
ソニー，東芝ドライブでもあまりに古いものでは音楽CDのアクセスはできませんが，最近低価格で出ているものなら問題はないでしょう。値段との兼ね合いもあるものの，これまでのSCSI機器におけるファームウェアの実績を考えれば東芝製のほうが若干信頼性が高いといえるかもしれません。どちらかでないと困る事態はないでしょうから，どちらでもかまいません。
それ以外のメーカーの製品ではどうでしょうか。現在のところ，先ほどの2社製品以外では音楽CDの読み込みはできませんし，CD-ROMドライバver.2.0に付属のツールはうまく動作しません。これらはソニー，東芝ドライブでもおかしくなることがありますので，むしろ計測技研のドライバ（もしくはライブラリ）に問題があるような気もします。
最近，低価格で3倍速，4倍速を発売しているNECドライブについては，ドライバの対応が完全でない，または相性が悪いということでMacFileTransfer.xやCDプレイヤーが動作しません。通常のファイルの読み込みやディレクトリの表示などといった基本的な部分ではなにも問題がありません。価格も安めなので，低価格で基本性能の高いドライブを求めている人にはよいかもしれません。
なお，Macintoshのディスクにアクセスできないとはいっても，それはSX-WINDOW上からMacFileTransfer.xを使った場合のことで，フリーソフトウェアのCDROM.Xを使用した場合は問題なくアクセスできます（使い勝手も格段によい)。要するにSX-WINDOWで使うかどうかで評価がまったく変わってくるでしょう。これは他社のドライブでも同様です。
ちなみにソニードライブでも最近2.4倍速のものが現れていますが（メルコのCDS-Eなど），これはモードによるもので高速アクセスモードコマンドは非公開になっているようです。X68000で普通に使用するときは倍速ドライブとなります。
次世代ゲーム機として扱われているもののひとつ，PC-FXがPC-9801にCD-ROMドライブとしてつながるということで少し気にしている人もいるかもしれません。長い間具体的なインタフェイスは明らかにされなかったのですが，最近になって，やはりSCSIであると発表されました。となればX68000との接続にもほとんど問題はないでしょう（若干不安もあるが……)。なにがしかの＋α分を考えれば49,800円という値段はまあまあ納得できるところではあります。少し問題になるのは，やはりNECドライブを採用しているのだろうなあというところでしょうか。
なお，友達からもらったとかいうのでもないかぎり，標準速のドライブには耐えがたいものがありますので，パスしたほうがよいでしょう。

4倍速は必要か？

今回取り上げたドライブはすべて倍速ドライブですが，市場にはすでに4倍速ドライブが増えつつあります。
4倍速ドライブならアクセス速度はともかく転送速度はハードディスク並になります（600Kバイト/秒）。しかし，X68000で使用する場合，それほど大量のデータを頻繁に処理することがあるとは考えにくいというのが実情です。
そのほか，同じ倍速でも，Macintoshならソニードライブでは滑らかに動作する動画が松下寿ドライブではガクガクするといった症状もあるようですが，X68000ではそれほどシビアに速度を要求されることはありませんからほとんどこだわる必要はありません。ついつい高速ドライブを選びたくなる気持ちもわかりますが，それに見あったメリットがあるかどうかは疑問です。
まずはドライバやツールと相性がよくて，低価格のものを選ぶべきでしょう。同じ金額を出すなら200Mバイト程度のワーク用ハードディスクを追加したほうが賢明かもしれません。

その他の要因

ドライブの機能/性能と値段以外にも製品選びのポイントはあります。それはデザインであったり，操作性であったりします。すなわち，トレイ式かキャディ式か，IDが固定されているのか可変なのか，電源は内蔵しているのかACアダプタなのか，電源スイッチは前面にあるのか背面にあるのか，どれくらいのスペースを占有するのか，といった問題です。
ディスクの入れ方で音楽CDのように前面から出てきたトレイにCDを置くのがトレイ式，CDキャディというカートリッジに入れて差し込むのがキャディ式です。最近は使いやすいトレイ式が増えています。今月紹介されたものはすべてトレイ式です。
IDはSCD-200以外は可変，電源はCDS-E以外は内蔵式，電源スイッチはCS-CD301Xは前面，それ以外は背面です。そのほか，今月紹介したものはすべてヘッドホン端子とオーディオOUTを持ち，ハーフピッチのSCSIコネクタを持ったものです。
ほかにもたくさんドライブは市販されています。各自の用途にあったポイントを押さえて選んでみてください。

各種CD-ROMを見る

巷にあふれるさまざまなデータ

CD-ROM

Nakano Shuichi 中野 修一

CD-ROMといっても，いったいどんなものが使えるのでしょうか？
有効に使えるデータというのは少ないものですが
ここではX68000でさまざまな機種のメディアを読んでみましょう

読めるもの読めないもの

世の中にはたくさんのCD-ROMがあります。基本的にCD-ROMというものはISO9660という規格において統一されています。しかし，必ずしもすべてがそれに従っているとは限らないのも事実です。

ここでは「読んでどうするのか？」というのは置いといて，読めるかどうかだけに焦点を絞ってさまざまな機種用のCD-ROMをアクセスしてみました。

●Macintosh

普通には読めません。計測技研のアクセスソフトまたはフリーソフトウェアのCDROM.Xが必要です。たまにまったく読めないものもあるかもしれません。

●MS-DOSマシン全般

基本的にすべて読めます。

●AMIGA（CDTV）

基本的にすべて読めます。特殊なファイル名がある場合は注意してください（途中にスペースを含んだものなど）。

●NeXT

基本的にすべて読めます。ソースつきのフリーソフトウェアも多いのですが，移植はちょっとしんどそうです。

●電子辞書（PCWING）

広辞苑その他の辞書類です。基本的にアクセス可能です。文字コードなどが特殊になっているのでそのままでは使えません。計測技研のSX広辞苑で使用可能です。

●PhotoCD

問題なく読めます（マルチセッション対応のドライブであること）。

●音楽CD

ドライブによってはCD2PCM.Xで生データにアクセス可能です。

●ビデオCD

ディレクトリなどは読めることは読めます。が，Human68kのファイル管理では超巨大なファイルがあるディレクトリ（およびファイル）はアクセスできないようです。MPEGのデータは30分のものでも270Mバイト以上になります。OS-9のMPEGボードでは再生可能。ちなみに，Human68k上でソフトウェアによりMPEGデータを再生するフリーソフトウェアとしてMPEGPLAY.Xがあります。

●電子ブック

基本的に読めます。やはり大きなファイル（辞書類）があるとアクセスできません。

●3DO

認識不可能です。

●プレイディア

基本的に読めません。ISO9660だと思われますが，ルートディレクトリに600Mバイト以上のファイルがあるのでHuman68kでは扱えません。

●MEGA-CD

問題なく読めます。

●PCエンジンROM²

基本的に読めません。ブロックダンプすればアクセスは不可能ではありません。

●NEO・GEO-CD

問題なく読めます。

●SATURN

問題なく読めます。

●PlayStation

問題なく読めます。

*　　　*　　　*

だいたいこんなところでしょう。CDIとOS/2用のものは確認しませんでしたが，まあ問題ないでしょう。現時点でこれ以外のものというとPC-8801MCとかJAGUARとかカーナビ用のCD-ROMくらいしか思い当たりません。これらについては入手できなかったので確認しませんでしたが，それ以外はわざわざX68000でこれらを読んでみようという人は少ないことと思います。

ざっと見てわかるように，CDROM.XやCD2PCM.Xといったフリーソフトウェアが非常に有効になってきます。これらは通信で入手するか，計測技研の「X68000フリーソフトウェアセレクション2」に収録されていますのでそちらで入手してください。

音楽CDの読み込みについて

本来，CD-ROMではオーディオCDの音声をデジタルデータとして読み込むことはできない設計になっていたのですが，最近ではその紳士協定もなくなってきているようです。

こういうのは著作権がらみの問題がやっかいなものなのですが，こういう機能が認められているからには著作権フリーなCDというのがどこかにあるのでしょうか。2,3年前にはレーザーディスクから綺麗にダビングできることを売り文句にしていたビデオデッキさえありましたし……。

ということで，X68000でもフリーソフトウェアのCD2PCM.Xというものを使えば，CDに録音されているデータをデジタル変換でX68000のAD PCM形式にすることも可能です。一般に市販されているCDにはほぼ間違いなく「複製を禁ずる」という表記があるはずですが，著作権フリーのCDがあったらぜひ試してみてください。まあ，以前「SPA!」誌に「レンタルCD屋で借りてダビングしたテープを友達にあげる」くらいのことは，著作権法上問題ないという解釈も載っていたので，個人で楽しむ分にはそれほど気にしなくてもよいのかもしれません（営利で使わないなら会社のソフトをコピーして自宅で使ってもかまわないと書いていた雑誌でもあるが……）。

さて，冗談はともかく，世間ではプログラム以外のデジタルデータに関する著作権については真面目に考えるのが馬鹿らしくなるほどいい加減です。

ラジオやテレビなどで使用される曲では音楽著作権料は支払われていません。実際

にはチェックする手間がかかりすぎるので徴収していないだけだそうです。そこでPCM放送についてデジタル技術を用いたチェック機構を組み込もうとしたところ，ある衛星TV局は既得権の侵害で営業妨害だと開き直ってきていますから（という記事を淡々と報道する新聞も凄いが），こういう業界では著作権感覚は微塵もないと考えていいでしょう。

ということで，現実には適当な判例もありませんし，扱いの難しい問題となっています。周りがいい加減だから自分もいい加減にしていいというわけでもありません。ただ一般的にいって，音楽CDについては1世代のみDATにデジタルコピーすることを認めるという線でコンセンサスがとられていますので，それに準拠した考え方でよいでしょう。あくまでも個人使用の範囲で，ですが。

そのほかのデジタルデータ

最近はライセンスフリーで使用できる素材CD-ROMもたくさん発売されています。こういったものについては，なにも問題はありません。商業目的に使用することも自由です。

また，一定の範囲内で自由に使用できることを明記したCD-ROMも結構あります。クリップアート集などでは，「そのデータを使ってクリップアート集を作らないこと」というものもありました。もっともな主張といえます。

ただ，そういったフリーデータの作者も知らないうちに他人の著作権を侵害していることもままあります。そういうのを見分けるのは不可能に近いことです。

参考までに，大企業ゆえにこういった問題を左右するであろうと思われるマイクロソフト社の判断ではデジタルデータの複製とは「メモリまたは外部記憶上に同じデータが同時に2つ以上存在すること」ということになっています。こうなるとディスクキャッシュも効きませんし，もうDSPでデジタルエフェクタなんか作れない世界ではあります。

日本での解釈はまだ流動的です。まあ特別な意思表示があるなら別ですが，常識的に考えて個人で楽しむ範囲でなら許容されると見るのが正解でしょう。

データの種類

●グラフィック

たいていは写真やスキャナからの取り込み画像です。マッピング素材などに使用できるものやアートクリップとして使えるものを探してみるのがよいでしょう。

壁紙用のものや単に眺めるだけのものではマル恥メディア的展開をしているものもたくさんあります。あまり効率はよくない気がしますが（深く追求しないこと），お好きな人はどうぞ。

●音楽

効果音サウンド関係は探せばそれなりのものがあります。MIDIデータはあまり期待してはいけません。「お，CD-ROM 2枚にスタンダードMIDIファイルが満載か！」と期待して買ってきても，聞かなきゃよかったと思うようなデータが満載されていることだってありますから。

●動画

Windowsのアプリなどで中にAVIファイルが入っていることがままあるのでそういったものを再生してみたいと思うこともあるでしょう。そういう場合に有効なフリーソフトウェアとしてQT_PLAY.X，AVIPLAY.Xがあります。これらはMacintoshなどのQuickTimeムービーやWindows AVIファイルを再生するためのソフトウェアです（要大メモリ）。ただし現在のところあまり大きなデータは再生できません。

●フリーソフトウェア

MacintoshやMS-DOS関係ではソースつきのフリーソフトウェアをまとめたCD-ROMもあります。はっきりいって使いものになりそうなツールは少ないのですが，言語関係ではC言語，C++，FORTRAN，PASCAL，MODURA2，ADAなどまで入っているものもありました。移植のお好きな方はどうぞ。

また，あまりフリーソフトウェアを入手する機会のない人なら，すでに発売されているX68000フリーソフトウェアセレクション1,2だけでもかなりの資産だとみなすこともできます。持っていて損はないでしょう。

●各種データ　テキストデータで膨大な情報を提供してくれるCD-ROMもあります。天文学的な恒星の軌道データを収録したり，日本全国の電話帳を打ち込んだものなどです。

*　*　*

CD-ROMタイトルを探すうえで大切なのは，なによりもまずCD-ROMの専門店を見つけることです。一般のパソコンショップで専門的なサポートを期待しても無駄なことは皆さんすでにご存じでしょう。

いずれにせよ，使える使えないという問題を除いても玉石混淆の状態ですので，当たりはずれは当たり前と思っていたほうが無難です。幸運を祈ります。では。

バックの鬼<寂>
真に使えるものだけを厳選したという背景素材集。多少疑問はあるが，まあ使える。

バックの鬼<響>
同シリーズの効果音版。動物の鳴き声やゲームで使える電子音なども収録。

MAC-SOURCE CD-ROM
Macintoshのフリーウェア集。言語などを多彩に揃える。ソースはあまり入ってない。

The Best of MIDI Connection
はずれの例。クラシックが数Kバイトなんでおかしいとは思ったんだ。標準MIDIファイル。

ENCYCLOPEDIA OF GRAPHICS FILE FORMATS
これは同名の書籍のおまけ。画像サンプルとソースつき。

黒船<全国版>
日本中の電話帳を手で打ち込んだという労作。59,800円と高価なのが難点か。CSVファイル。

CD-ROM

画像ローダの作成

PhotoCDデータの活用

Kikuchi Isao 菊地 功

もっとも手軽に作成できるCD-ROM，PhotoCD
それは高品位なグラフィックデータを提供してくれます
ここではデータを活用する高画質画像ローダを作りましょう

CD-ROMドライブの普及とともに最近ではデータの配布媒体としてCD-ROMがかなり幅をきかせつつあります。

CD-ROMのメリットは容量当たりのコストの安さにあります。これまで配布媒体として広く利用されてきたフロッピーディスクに比べ，約500倍の記憶容量を持ち，コストはフロッピー2枚分より安い（中身のことを考えなければですが）というのですから，大容量時代にはまさにもってこいのメディアといえるでしょう。

雑誌の付録などにもよく利用され（その内容には疑問を感じるものも多数ありますが），ソフトもフロッピーディスク版よりもCD-ROM版のほうが安いといったことがままあります。近い将来はほとんどがCD-ROMで配布されるようになるのでしょう（フロッピーディスクも残るでしょうが）。

さて，このCD-ROMですが，残念ながらX680x0の世界では，まだそれほど普及しているとはいえないようです（普及率でいえばMOのほうが高いかも）。そもそもX680x0で利用できるCD-ROMがほとんどないのが現状ですから，しかたのないことかもしれません（私もCD-ROMドライブを保有していますがX68000には繋がっていません）。

これがPhoto CDだ

だからといって諦めてしまうには，あまりにももったいないので，今回はPhotoCDの利用法について考えてみました。

PhotoCDとは

本誌1993年11月号でも紹介されていますが，PhotoCDとはコダック社が開発したCD-ROMフォーマットであり，35mmフィルムからスキャナで画像を起こしてライトワンスCDに書き込んだものです。PhotoCDには100枚までの画像を収めることができます。100枚に満たないPhotoCDがある場合にはそれに追加書き込みをすることもできて経済的です（追加書き込みをしたPhotoCDにアクセスするには，マルチセッションに対応したCD-ROMドライブが必要です）。

PhotoCDを焼くには，PhotoCDを扱っているお店に未現像フィルムもしくは現像済みネガ（またはポジ）を持っていって，「PhotoCDに焼いてちょ」とお願いします。

基本料金が500円，メディア代が1000円，さらに写真1枚書き込むたびに未現像フィルムなら80円，現像済みネガなら100円の料金がかかります。たとえば36枚撮った未現像フィルムからPhotoCDを焼く場合なら，500＋1000＋80×36＝4380円となります。画質と保存を考えた場合，まあ妥当な値段なんじゃないでしょうか。

また，PhotoCD以外にも普通のCD-ROMソフトの中にPhotoCDデータ（拡張子.PCD）が入っていることもあります。写真をPhotoCDで取っておこうという人以外でもPhotoCDのデータに出食わすことがあるわけです。

で，このPhotoCDデータを見る方法ですが，計測技研からX680x0用のビュアでSX-PhotoGalleryという製品が発売されています。しかし，このビュアはSX-WINDOW専用ですし，結構いい値段しますので，もっと手軽に観賞する方法について考えてみましょう。

フォーマット

おそらく世にあるほとんどのPhotoCDビュアはPhotoCDのセクタから直接読み出しているものと思われます。しかし，PhotoCDはISO9660に準拠しており，ディレクトリを取ると，ちゃんと画像データがファイルになっていることがわかります。そこで，このファイルから画像を抽出することを試みました。それぞれの画像が収められたファイルは，\PHOTO_CD\IMAGESのディレクトリ内にあるIMG????.PCD（????は通し番号）です。

100枚までなのになんで4桁なのかはどうでもいいとして，ひとつのファイルの中に192×128，384×256，768×512のサイズの画像が収められています。実際にはもっと大きな画像も収められているはずですが，上の3つとは違うフォーマットで収められているようなので，諦めました（上の3つは比較的簡単なフォーマットで収められています）。画面に表示することを前提にすれば，768×512が読めればほとんど問題ないでしょう。

さて，そのフォーマットですが，PhotoCDはコダックYCC方式で記録されています。テレビやビデオについて，よく「YC分離」という言葉を耳にしますが，あれと同じようなものです。Yは輝度信号，Cは色信号ですが，PhotoCDでは赤と青の色差信号CrとCbで記録されています。もともとはテレビに表示することを目的にしていたので，このようなフォーマットになっているのでしょうが，パソコンで表示するためにはこれをRGBに変換しなければなりません。一般的にYCrCbからRGBへの変換式は，

$$R = Y + 1.40200 \times Cr$$

G＝Y－0.34414×Cb－0.71414×Cr

B＝Y＋1.77200×Cb

となるのですが，PhotoCDでは補正などがあるようで，次のようになっています。

R＝(1391×Y＋1865×Cr－255023)÷1024

G＝(1391×Y－441×Cb－949×Cr＋199313)÷1024

B＝(1391×Y＋2271×Cb－353784)÷1024

ただし，これはRGB各8ビットのフルカラーの話ですので，X680x0の65536色モードで表示するにはRGBをそれぞれ8で割って5ビットにしてやる必要があります。

それぞれのデータの格納位置ですが，画像サイズが192×128，384×256，768×512のデータはそれぞれファイルの先頭から$2000～$b000，$b800～$2f800，$30000～$c0000に格納されています。先頭から$1fffまでや，それぞれの間の空きにはたいした情報は含まれていないようですので，無視していいでしょう。

さらにそれぞれの領域でのYCCの格納法を図1に示します。ここでは192×128のサイズの画像の格納方法を示しましたが，384×256や768×512でもドット数が違うだけで，基本的な構造は同じです。

Y,Cr,Cbはそれぞれ1バイトで扱われていますが，ここで注意が必要なのはCbとCrが2×2ピクセルにひとつしか存在しないことです。テレビなどでも色信号がずさんに扱われるように，PhotoCDも自然画である限り，輝度信号をちゃんと保存しておけば色信号は1ドット飛ばしても不自然はないと判断したのでしょう。実際，ちゃんと補間してやれば問題になることはまずないと思われます。

表示してみる

フォーマットがわかったところで，さっそく表示してみましょう。画面モードは当然65536色モードを使用します。画像サイズは，できるだけ大きくしておけばあとはどうにでもなりますので，768×512を使用することにします。X680x0では65536色は512×512ドットしか描画できませんが，アスペクト比を合わせるために横768ドットを間引いて512ドットにすることにします。ただし，ドットをそのまま表示してほしいという場合もあるかもしれませんので，オプション指定で横768ドットのうちの任意の512ドットを切り出せるようにもしましょう。

プログラムをリスト1に示します。

YCCからRGBへの変換は式に放り込んでしまえばそれまでなので，たいした問題ではないでしょう。それよりもプログラムのポイントは色信号の補間です。Y方向への補間の必要性から，Cb,Crのバッファを2本取って先読みさせ，その関係でYのバッファも4本取っています。

補間の方法としては，単純平均をとっているだけですが，画像の右端や最下ラインは平均を取る相手がいませんので，それぞれすぐ左および上の色信号をそのまま使用しています。また，元データはフルカラーということで，2×2の単純ディザを掛けて表示しています。91～93行のYCC→RGB変換時に行っており，これによってマッハバンドの出現を防ぐことができますが，画像によっては縞模様に見えてしまうことがあります。どうしても気になる方は，

(i<<12)＋(j<<11)

の部分を4096（四捨五入のため）にしてみてください。ディザを掛けないようになります（マッハバンドは出ますが）。

さて，画面に表示できるようになりました。しかし，観賞するだけならともかく，加工したいとなると，やっぱりZ's STAFFやMATIERから読み出したいところですよね。そこで，例によってEX-Windowの登

図1　192×128ドットの実際のデータ構造

$2000	Yのデータが192個（192BYTE）	1ライン目のY
+192	Yのデータが192個（192BYTE）	2ライン目のY
+384	Cbのデータが96個（96BYTE）	1ライン目のCb（1ドット飛ばし）
+480	Crのデータが96個（96BYTE）	1ライン目のCr（1ドット飛ばし）
+576	Yのデータが192個（192BYTE）	3ライン目のY
+768	Yのデータが192個（192BYTE）	4ライン目のY
+960	Cbのデータが96個（96BYTE）	3ライン目のCb（1ドット飛ばし）
+1056	Crのデータが96個（96BYTE）	3ライン目のCr（1ドット飛ばし）
+1152	Yのデータが192個（192BYTE）	5ライン目のY
	⋮	

リスト1

```
 1: #include       <stdlib.h>
 2: #include       <stdio.h>
 3: #include       <string.h>
 4: #include       <doslib.h>
 5: #include       <iocslib.h>
 6:
 7: #define MAX_X   768
 8:
 9: void    usage();
10:
11: unsigned char   Y[4][MAX_X];
12: unsigned char   Cb[2][MAX_X/2];
13: unsigned char   Cr[2][MAX_X/2];
14:
15: void    main( int ac, char *av[] )
16: {
17:         FILE    *fp;
18:         char    filename[90] = { 0 };
19:         int     i, j, x, y, k1, k2;
20:         int     x0=0, Loption=0;
21:         int     xsize, ysize, dx;
22:         long    offset;
23:         int     r, g, b, cb, cr;
24:         unsigned short  *vp = (unsigned short *)0xC00000;
25:
26:         xsize = 768;
27:         ysize = 512;
28:         dx = 768;
29:         offset = 0x30000;
30:         CRTMOD( 12 );
31:         G_CLR_ON();
32:         printf( "Kodak PhotoCD PCD Loader¥t(c)Isawo-Kikuchi¥n" );
33:         for( i=1; i<ac; i++ ){
34:           if( av[i][0]=='/' || av[i][0]=='-' ){
35:             if( av[i][1]|0x20=='l' ){
36:               Loption = 1;
37:               x0 = atoi( &av[i][2] );
38:               dx = 512;
39:               if( x0<0 || x0>256 ){
40:                 usage();
41:                 return;
42:               }
43:             } else {
```

まあ，こういうのもありがち

ジャンルは多彩だ

場です。といっても，外部ファイルを作成したりする必要はありません。5月号のこいのぼりPRO-68KにIMGFILERという便利なものが収録されているじゃありませんか。説明は5月号に譲るとして，コンフィグファイルに，

:PCDFiler
　　0，0
　　IMGFILER PCDL "" "" PCD

と書き加えるだけで，EX-WindowからPCDファイルをロードできるようになります（もちろん，こいのぼりPRO-68Kに掲載されたIMGFILER.Xと，リスト1をコンパイルしてできたPCDL.Xをパスの通ったディレクトリに置いておかなければなりませんが）。

変換してみる

画面に表示してしまうと，VRAMの関係でどうしても間引かれたり，部分切り出しになってしまいます。しかし，元データをできるだけ傷つけずに保存したい，という場合もあるかもしれません。そこで今度は，MATIERやSX-WINDOW（？）でサポートしているGLMファイルに変換してみましょう。

それにはまず，GLMのフォーマットを知る必要があります。といっても，べたデータに図2のような16バイトのヘッダがついているだけです。グラフィックデータは，緑5ビット，赤5ビット，青5ビット，輝度1ビットの順に1ピクセル2バイトで格

```
 44:                usage();
 45:                return;
 46:              }
 47:            } else strmfe( filename, av[i], "PCD" );
 48:          }
 49:          if( filename[0]==0 ){
 50:            usage();
 51:            return;
 52:          }
 53:          fp = fopen( filename, "rb" );
 54:          if( fp==NULL ){
 55:            printf( "%s が見つかりません。¥n", filename );
 56:            return;
 57:          }
 58:          if( fseek( fp, offset, 0 ) ){
 59:            printf( "ファイルシークエラーです。¥n" );
 60:            return;
 61:          }
 62:          SUPER( 0 );
 63:          for( i=0; i<2; i++ ) fread( Y[i], 1, xsize, fp );
 64:          fread( Cb[0], 1, xsize/2, fp );
 65:          fread( Cr[0], 1, xsize/2, fp );
 66:          for( y=0; y<ysize/2; y++ ){
 67:            k1 = y%2;
 68:            k2 = (y+1)%2;
 69:            for( i=0; i<2; i++ ) fread( Y[k2*2+i], 1, xsize, fp );
 70:            fread( Cb[k2], 1, xsize/2, fp );
 71:            fread( Cr[k2], 1, xsize/2, fp );
 72:            for( i=0; i<2; i++ ){
 73:              for( x=x0, j=x0%2; x<x0+dx; x++ ){
 74:                if( Loption==0 && x%3==2 ) continue;
 75:                if( (x%2)==0 || x==xsize-1 ){
 76:                  if( i==0 || y==ysize/2-1 ){
 77:                    cb = Cb[k1][x/2]; cr = Cr[k1][x/2];
 78:                  } else {
 79:                    cb = (Cb[k1][x/2]+Cb[k2][x/2]+1)>>1;
 80:                    cr = (Cr[k1][x/2]+Cr[k2][x/2]+1)>>1;
 81:                  }
 82:                } else {
 83:                  if( i==0 || y==ysize/2-1 ){
 84:                    cb = (Cb[k1][x/2]+Cb[k1][x/2+1]+1)>>1;
 85:                    cr = (Cr[k1][x/2]+Cr[k1][x/2+1]+1)>>1;
 86:                  } else {
 87:                    cb = (Cb[k1][x/2]+Cb[k2][x/2]+Cb[k1][x/2+1]+Cb[k2][x/2+1]+2)>>2;
 88:                    cr = (Cr[k1][x/2]+Cr[k2][x/2]+Cr[k1][x/2+1]+Cr[k2][x/2+1]+2)>>2;
 89:                  }
 90:                }
 91:                r = (1391*Y[k1*2+i][x]+1865*cr-255023+(i<<12)+(j<<11))>>13;
 92:                g = (1391*Y[k1*2+i][x]-441*cb-949*cr+199313+(i<<12)+(j<<11))>>13;
 93:                b = (1391*Y[k1*2+i][x]+2271*cb-353784+(i<<12)+(j<<11))>>13;
 94:                if( r<0 ) r = 0;
 95:                if( g<0 ) g = 0;
 96:                if( b<0 ) b = 0;
 97:                if( r>31 ) r = 31;
 98:                if( g>31 ) g = 31;
 99:                if( b>31 ) b = 31;
100:                *(vp++) = (g<<11)|(r<<6)|(b<<1);
101:                j ^= 1;
102:              }
103:            }
104:          }
105:          fclose( fp );
106: }
107:
108: void     usage()
109: {
110:          printf( "[使用法] PCDL [/L[X]] <filename>¥n"
111:            "PCDファイルから768×512の画像を512×512に間引いて表示します。¥n"
112:            "/Lオプションを指定すると、間引きを行わず、指定されたX座標から¥n"
113:            "表示します。( 0 ≦ X ≦ 256 )¥n"
114:          );
115: }
```

リスト2

```
  1: #include       <stdlib.h>
  2: #include       <stdio.h>
  3: #include       <string.h>
  4:
  5: #define MAX_X  768
  6:
  7: void     usage();
  8:
  9: unsigned char   Y[4][MAX_X];
 10: unsigned char   Cb[2][MAX_X/2];
 11: unsigned char   Cr[2][MAX_X/2];
 12:
 13: void     main( int ac, char *av[] )
 14: {
 15:          FILE    *fp1, *fp2;
 16:          char    pcdname[90], glmname[90];
 17:          int     i, j, x, y, k1, k2;
```

納されています。

早い話が，VRAMそのままですね。というわけで，プログラムはPCDL.Cの書き出し先をファイルにした程度ですので，特に説明することもないでしょう。

さて，GLMへの変換ができました。が，実は生成されたGLMファイルは，SX-WINDOWでは表示できません。どうやらSX-WINDOWでは，512×512より大きな画像はちゃんとロードできないようです。その辺は私の責任じゃないのでほっといて，念のためMATIERでの使用法を示しておきましょう。普通どおりに表画面に部分ロードしてもいいのですが，せっかくなので仮想画面を使って，全画面ロードしましょう。

まず，LUPEの上にあるVIRTUALの4つのボタンのうち，どれでもいいからクリックしてみてください。メモリが十分あれば，仮想画面ウィンドウが開くはずです。そこで横サイズを768以上にしてOKをクリックし，仮想画面を確保してください。次にファイルメニューにいき，画像形式をGLM，対象画面を仮想画面にして，PCD2GLMで生成されたファイルをロードします。これで仮想画面に全体が読み込まれたはずですので，あとは煮るなり焼くなり好きなようにしてください。

その他

今回は768×512のサイズのデータしか扱いませんでしたが，192×128や384×256のデータも，ソースをちょっといじるだけで対応させることができます。

また，¥PHOTO_CD¥OVERVIEW.PCDには，同じ構造で$2800から192×128のデータが枚数分詰めて格納されているようですので，インデックスなどを作る際には，そちらを利用すると便利かもしれません。

私もPhotoCDを焼いてみたいんですが，私自身写真を撮らない人間なので，焼くものがないんですよね。PhotoCDを焼くために写真を撮るか，それとも誰かに撮らせるか。それが問題です。

図2 GLMのフォーマット

$0000～$0003	識別子（'G','R','6','5'）
$0004～$0007	ファイルエンドまでのバイト数（ファイルサイズ－8）
$0008～$000B	0に固定（たぶん）
$000C～$000D	Xサイズ
$000E～$000F	Yサイズ
$0010～	グラフィックデータ

```
 18:          int     xsize, ysize;
 19:          long    offset;
 20:          int     r, g, b, cb, cr;
 21:          unsigned short  dat;
 22:
 23:          xsize = 768;
 24:          ysize = 512;
 25:          offset = 0x30000;
 26:          printf( "Kodak PhotoCD PCD to GLM Converter¥t(c)Isawo-Kikuchi¥n" );
 27:          if( ac>=2 && ac<=3 ){
 28:            strmfe( pcdname, av[1], "PCD" );
 29:            if( ac==2 ) strmfe( glmname, av[1], "GLM" );
 30:            else strmfe( glmname, av[2], "GLM" );
 31:          } else {
 32:            usage();
 33:            return;
 34:          }
 35:          fp1 = fopen( pcdname, "rb" );
 36:          if( fp1==NULL ){
 37:            printf( "%s が見つかりません。¥n", pcdname );
 38:            return;
 39:          }
 40:          if( fseek( fp1, offset, 0 ) ){
 41:            printf( "ファイルシークエラーです。¥n" );
 42:            return;
 43:          }
 44:          fp2 = fopen( glmname, "wb" );
 45:          if( fp2==NULL ){
 46:            printf( "%sを作れません。¥n", glmname );
 47:            fclose( fp1 );
 48:            return;
 49:          }
 50:          fputc( 'G', fp2 ); fputc( 'R', fp2 ); fputc( '6', fp2 ); fputc( '5', fp2 );
 51:          i = 768*512*2+8;
 52:          fwrite( &i, sizeof( int ), 1, fp2 );
 53:          fputc( 0, fp2 ); fputc( 0, fp2 ); fputc( 0, fp2 ); fputc( 0, fp2 );
 54:          fputc( xsize>>8, fp2 ); fputc( xsize, fp2 );
 55:          fputc( ysize>>8, fp2 ); fputc( ysize, fp2 );
 56:          for( i=0; i<2; i++ ) fread( Y[i], 1, xsize, fp1 );
 57:          fread( Cb[0], 1, xsize/2, fp1 );
 58:          fread( Cr[0], 1, xsize/2, fp1 );
 59:          for( y=0; y<ysize/2; y++ ){
 60:            k1 = y%2;
 61:            k2 = (y+1)%2;
 62:            for( i=0; i<2; i++ ) fread( Y[k2*2+i], 1, xsize, fp1 );
 63:            fread( Cb[k2], 1, xsize/2, fp1 );
 64:            fread( Cr[k2], 1, xsize/2, fp1 );
 65:            for( i=0; i<2; i++ ){
 66:              for( x=0, j=0; x<xsize; x++ ){
 67:                if( (x%2)==0 || x==xsize-1 ){
 68:                  if( i==0 || y==ysize/2-1 ){
 69:                    cb = Cb[k1][x/2]; cr = Cr[k1][x/2];
 70:                  } else {
 71:                    cb = (Cb[k1][x/2]+Cb[k2][x/2]+1)>>1;
 72:                    cr = (Cr[k1][x/2]+Cr[k2][x/2]+1)>>1;
 73:                  }
 74:                } else {
 75:                  if( i==0 || y==ysize/2-1 ){
 76:                    cb = (Cb[k1][x/2]+Cb[k1][x/2+1]+1)>>1;
 77:                    cr = (Cr[k1][x/2]+Cr[k1][x/2+1]+1)>>1;
 78:                  } else {
 79:                    cb = (Cb[k1][x/2]+Cb[k2][x/2]+Cb[k1][x/2+1]+Cb[k2][x/2+1]+2)>>2;
 80:                    cr = (Cr[k1][x/2]+Cr[k2][x/2]+Cr[k1][x/2+1]+Cr[k2][x/2+1]+2)>>2;
 81:                  }
 82:                }
 83:                r = (1391*Y[k1*2+i][x]+1865*cr-255023+(i<<12)+(j<<11))>>13;
 84:                g = (1391*Y[k1*2+i][x]-441*cb-949*cr+199313+(i<<12)+(j<<11))>>13;
 85:                b = (1391*Y[k1*2+i][x]+2271*cb-353784+(i<<12)+(j<<11))>>13;
 86:                if( r<0 ) r = 0;
 87:                if( g<0 ) g = 0;
 88:                if( b<0 ) b = 0;
 89:                if( r>31 ) r = 31;
 90:                if( g>31 ) g = 31;
 91:                if( b>31 ) b = 31;
 92:                dat = (g<<11)|(r<<6)|(b<<1);
 93:                fwrite( &dat, sizeof( short ), 1, fp2 );
 94:                j ^= 1;
 95:              }
 96:            }
 97:          }
 98:          fclose( fp1 );
 99:          fclose( fp2 );
100: }
101:
102: void     usage()
103: {
104:          printf( "[使用法] PCD2GLM <PCDname> [<GLMname>]¥n"
105:            "PCDファイルから768×512の画像をGLMファイルに変換します。¥n"
106:            "GLMファイル名を省略すると、PCDベース名.GLM になります。¥n"
107:          );
108: }
```

CD-ROM

SXSLICE.Xの作成

BMPファイルを表示する

Ishigami Tatsuya 石上 達也

最近のCD-ROM上でのグラフィックデータ形式の代表例が
Windowsの標準グラフィックファイルBMPファイルです
ここではSX-WINDOWでBMPファイルを表示するためのプログラムを作成しました

最近ではCD-ROMソフトがずいぶん揃ってきました。X68000で直接使えるものは少ないのですが，たいていのものは，その容量を生かして大量のデータを収録しています。そういったデータのなかで，X68000でも比較的簡単に利用できるものにグラフィックデータがあります。

従来，CD-ROMといえばMacintosh用のものが多かったのですが，特に最近幅をきかせているのがWindows用のCD-ROMです。これはCD-ROMドライバがあればX68000でも直接読めますし，入手も簡単です。この記事ではグラフィックデータ形式のなかでも主にマイクロソフト社のOSで使われているDIB形式，通称BMPファイルと呼ばれているものについて説明します。具体的には1994年8月号で発表したSX-picsliceをBMPファイル対応にしました。対応ファイルがPICだけではなくなったので，名前もSX-sliceと改めました。

グラフィック画面の表示

SX-WINDOWでは，コントロールパネルから「背景選択」を実行することにより，壁紙を自由に変更することができます。

普通は方眼模様やタイル模様を壁紙として使用するのですが，このときPAT4形式のグラフィックデータを使用することができます（パターンエディタで表示→「コピー」メニューでクリップボードへ転送→「背景選択」へペースト）。

BMPを壁紙に

で，なぜだかは知らないのですが，ここで16×16ドットの大きさの透明なパターンを背景として用いると，背景（テキスト画面）が透けてグラフィック画面が表示されるようになります。

この作業を行っておけば，グラフィック画面を壁紙として使用することができるようになるわけです。SX-picsliceは直接G-RAMにPICデータを表示するツールでしたが，今回はBMPも表示できるように拡張するわけです。

使い方

掲載されたリストと「もみじ狩りPRO-68K」に収録されているSX-picsliceのソースからpicslice.c，picslice.hを抜き出してmakeすれば実行ファイルができあがります。

プログラムの実行には，SX-picsliceと同様にコマンドラインからパラメータを与える方法と，SX-BASICからタスク間通信により命令する方法の2種類あります。以下に，必要な部分を再掲載します。

●コマンドラインから

-fファイル名

与えられたファイル名で示されるファイルを表示します。

例）id=fock("sxpics.x -fcc.pic")

-v0

起動後，SX-sliceのウィンドウを開きません。

-r数値

SX-sliceは，一定幅のラスターを処理するごとに制御をSX-WINDOWに返しますが，その幅を指定します。ここに512を指定すると1画面分一度に展開するようになります。

例）id = fock("sxpics.x -r10")

●SX-BASICから

FILE ファイル名

与えられたファイル名で示されるファイルを表示します。

例）int id
　id=fock("sxpics.x")
　sendmes(id, "FILE cc.pic")

SHOW

SX-sliceのウィンドウを表示します。すでに表示されている場合はなにも起こりません。

HIDE

SX-sliceのウィンドウを隠します。すでに隠れている場合には，なにも起こりません。

RASTER

SX-sliceは，一定幅のラスターを処理するごとに制御をSX-WINDOWに返しますが，その幅を指定します。ここに512を指定すると1画面分一度に展開するようになります。

QUIT もしくは END

SX-sliceを終了します。

*　　*　　*

ついでに複数のSX-sliceを起動した場合の動作について解説します。

SX-sliceは，X68000のグラフィック画面を，ウィンドウ情報に関係なく1枚の描画領域として扱います。描画領域がひとつしかないので，SX-sliceが複数あっても意味がありません（FM音源がひとつしかないのにサウンドプレイヤーが複数あっても意味がないのと同様）。

2つ目以降のSX-sliceは，

・コマンドラインにパラメータが与えられていなかった場合
　→直ちに終了。

・コマンドラインにパラメータが与えられていた場合
　→すでに起動されているSX-sliceに対し，タスク間通信を用いて，自分に与えられた

パラメータを託して，終了。
というように動作します。

BMPファイルについて

X68000ユーザーのなかには，BMPファイルと聞いてもあまりピンとこない方もいるかもしれません。

BMPファイルとはマイクロソフト社の制定したグラフィックデータのフォーマットで，主にWindowsやOS/2でグラフィックを扱う場合に使用されます。

Windows ver.1.0（発表はIBM版が1985年7月，PC-9801版が1987年6月）からすでに採用されており，長い歴史と共に，いくつかのバージョンが存在します（分類は参考文献1による）。

●OS/2 ver.1.x用

グラフィックの物理的なデータ（縦，横の大きさ，使用プレーン，1ドットあたりの使用バイト数）しか持っていない。よって，異なるアスペクト比（縦横比）を持つディスプレイ間での可搬性がない。また，データの圧縮を行わない。

●Windows ver.3.x用

OS/2 ver.1.x用に比べ，ディスプレイのアスペクト比を表すデータ（1mあたり，何ドット存在するか），使用色数，その色のなかでも重要な色の数を表すデータが追加されている。このため，アスペクト比や，同時発光可能色数の異なるディスプレイ間でもデータの共有が可能。また，ランレングス方式のアルゴリズムが採用され，データの圧縮が可能となっている。

●OS/2 ver.2.x用

Windows ver.3.x用に比べ，ディスプレイの分解能を表す数値の単位が選べるようになった，中間色を表すアルゴリズムが選べるようになった，アプリケーションが独自に使用できる変数領域が予約された，などの拡張が施されている。

と，以上のような種類のフォーマットがあるのですが，すべてサポートするのは大変ですし，現在ではほとんど使われなくなったものもあります。そこで今回は，

Windows3.x用
非圧縮

のもののみをサポートすることにしました。以上の条件を満たせば，16色データだろうが，16777216色データ（$=2^{24}$。通常はTrue colorと呼ばれています）だろうが，どのようなサイズのデータだろうが，すべて自動的に拡大/縮小，減色処理を行ったうえで512×512ドット，32768色データに変換し表示します。

よほど特殊なものでない限り，たいていのBMPファイルは表示できるはずです。

BMPファイルのフォーマット

BMPフォーマットは，大まかに見て，

File Header
Bitmap Information Header
Image Data

の3要素から構成されています。

●File Header

ファイルそのものに関するデータです。C言語の構造体で記述すると，以下のようになります。

```
typedef struct
        tagBITMAPFILEHEADER {
    WORD        Type;
    DWORD       Size;
    WORD        Reserved1;
    WORD        Reserved2;
    DWORD       OffBits;
  } BITMAPFILEHEADER
```

Type

Windows 1.x/2.x用のBMPファイルでは常に0，Windows 3.x/OS/2用のBMPファイルでは，BMPファイルであることを表す0x4d42（上位と下位をひっくり返して，ASCIIコードで表すと「BM」）が入っています。現在，流通しているBMPファイルのすべてが後者であると考えてよいでしょう（SX-sliceも前者については，サポートしていません）。

Size

ファイルの物理的な大きさが入っています。コマンドシェルからdirしたときに表示される大きさと同じものです。

Reserved1，Reserved2

拡張用に予約されています。現在，値は常に0でなければなりません。

OffBits

ファイルの先頭からImage Dataへのオフセットが収められています。

●Bitmap Information Header

ビットマップ情報に関する補助的なデータです。うまく活用すればデバイスに非依存な環境を構築できます。ただし，X68000の動作環境が何種類もないので，SX-sliceではあまり利用していません。

また，このデータはWindows 3.x用とOS/2用では若干異なっており，プログラムで区別が必要です。このデータを用いたデバイス非依存なBMPファイルを特にDIB(Device Independent BMP)というようです．

・Windows 3.0用

```
typedef struct
        tagBITMAPINFOHEADER {
    DWORD       Size;
    DWORD       Width;
    DWORD       Height;
    WORD        Planes;
    WORD        BitCount;
    DWORD       Compression;
    DWORD       SizeImage;
    DWORD       XPelsPerMeter;
    DWORD       XPelsPerMeter;
    DWORD       ClrUsed;
    DWORD       ClrImportant;
  } BITMAPINFOHEADER;
```

・OS/2 1.x用

```
typedef struct
        tagBITMAPINFOHEADER {
    DWORD       Size;
    WORD        Width;
    WORD        Height;
    WORD        Planes;
  } BITMAPINFOHEADER;
```

・OS/2 2.x用

```
typedef struct
        tagBITMAPINFOHEADER {
    DWORD       Size;
    WORD        Width;
    WORD        Height;
    WORD        Planes;
    WORD        BitCount;
    DWORD       Compression;
    DWORD       SizeImage;
    DWORD       XPelsPerMeter;
    DWORD       YPelsPerMeter;
    DWORD       ClrUsed;
    DWORD       ClrImportant;
    WORD        Units;
    WORD        Reserved;
    WORD        Rendering;
    DWORD       Size1;
    DWORD       Size2;
    DWORD       ColorEncoding;
    DWORD       Identifier;
  } BITMAPINFOHEADER;
```

Size

BITMAPINFOHEADERの大きさが入っています。この値でBMPファイルがどのOS用であるかを判断します。SizeとOSとの対応は以下のとおりです。

```
Size = 12    OS/2 1.x
Size = 40    Windows 3.x
Size = 64    OS/2 2.x
```

Width, Height

表示しようとするグラフィックデータの(幅，高さ)がドット単位で収められています。WindowsとOS/2では，このデータを表すのに必要なバイト数が異なるので注意が必要です。

Planes

表示しようとするグラフィックデータの使用プレーン数が収められています。

SX-sliceではplane=1以外のグラフィックデータ（複数のグラフィックデータの重ね合わせにより構成されているもの）はサポートしていません。

BitCount

1ドットを表すのに何ビットを使用しているかを表します。具体的には，

BitCount = 1	2色（白黒画像）
BitCount = 4	16色
BitCount = 8	256色
BitCount = 24	16777216色

となっています。

Compression

使用している圧縮アルゴリズムです。

Compression = 0	非圧縮
Compression = 1	RLE8
Compression = 2	RLE4

となっています。アルゴリズムの具体的な内容については後述します。SX-sliceではCompression=0のファイルのみサポートしています。

SizeImage

Image Dataの大きさが入っています。

XPelsPerMeter, YPelsPerMeter

ディスプレイのX，Y方向の分解能が入っています。上手に使うと，デバイス非依存なデータの活用ができます。

ClrUsed

必要な色変換テーブルの大きさが入っています。

……はずですが，手元にあったファイルでは，ほとんどのものが，ここに0を代入していました。SX-sliceでは，

clrUsed=(offBits−Size−14)/4

とすることにより，色変換テーブルの大きさを求めています（14=sizeof(BITMAPFILEHEADER), 4 = sizeof(RGBQUAD)です)。

ClrUsed= 0の場合，Image DataはB,G,Rの順にそのまま入っています。色変換テーブルは必要ありません。

ClrUsed≠ 0の場合, ClrUsedに代入された数だけ色変換テーブルを用意しなければなりません。

ClrImportant

使用されている色のなかでも，特に重要な色の数を示しています。16色しか発色できないビデオカードで256色のデータを表示する場合などに用いられるのではないかと思われます。X68000のグラフィック画面は，R，G，Bそれぞれの使用データ数に偏りがないので，SX-sliceではこの項を無視しています。

Units

分解能を表す数値単位のようです。詳細はわかりません。

Reserved

将来の拡張に備えて予約されています。現在のところ，常に0です。

Rendering, Size1, Size2

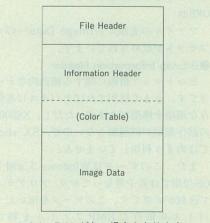
BMPファイルでハーフトーンを用いるときに使用されるようです。詳細はわかりません。

ColorEncoding

カラーテーブルのフォーマットを表しています。現在，常に0のようです。

Identifier

アプリケーションが自由に使用してよいことになっているようです。

●Color Table

BitCount≠24，かつ，ClrUsed≠ 0の場合，Bitmap Informationの直後に色変換用のカラーテーブルが収められています。

・OS/2 1.x用

```
typedef struct tagRGBTRIPLE
{
    BYTE  Blue;
    BYTE  Green;
    BYTE  Red;
} RGBTRIPLE;
```

・Windows 3.x, OS/2 2.x用

```
typedef struct tagRGBQUAD
{
    BYTE  Blue;
    BYTE  Green;
    BYTE  Red;
    BYTE  Reserved;
} RGBQUAD;
```

いちばん初めにくるテーブルがカラーNo=0で示される色のR, G, B成分ということになります。以下，カラーNo=1，2，……と収められ，ClrUsedで示された色数分続きます。

ちなみに，ここでのReservedには常に0が収められているようです。

●Image Data

カラー変換テーブルがない場合はBlue, Green, Redの順に1バイトずつ収められていますので，そのまま1ドットずつ表示していきます。

カラー変換テーブルがある場合はBitCountビット=1ドットですので，その値に相当するR, G, B成分をカラー変換テーブルにより求め，その色を表示します。

図1　BMPファイルの走査の方法

※一般の（x，y）座標系とはy方向の取り方が逆になります。

図2　BMPファイルの構成

※Color Tableがない場合もあります。

プログラムについて

BMPファイルはそんなに複雑なフォーマットではありませんし，そのローダを再利用する場面がそう多くあるとは思えません。

SX-picsliceのように独立したライブラリとするのが理想的ですが，今回は，SX-picsliceと使用変数をある程度共有化するにとどめています。

共有している変数は，picslice.h中で定義された構造体_picsliceinfoは以下のとおりです。

filehandle	ファイルハンドル
filebuf	読み込みバッファ
filebufsize	filebufの大きさ
currentbyte	filebuf上のREAD位置

ファイルの読み込みに関しては，先頭から必要な分だけを取ってくるようにしてメ

モリ使用量を最小限に抑えるのがslice の主旨なのですが，一部のMOやCD-ROMのように，細切れのシーケンシャルリードに対してあまりスピードが期待できないデバイスを考慮し，あえてそのような方式をとっていません。ファイルをオープンすると，一気にメモリ上のバッファに読み込み，必要なデータはそのバッファから読んで表示が終わったらバッファを破棄するという方法をとっています。

おまけ

SX-sliceの制作に先立って，私の作ったヘッダ部分を解析するプログラムも紹介します。MATIERの現バージョンでは，まだBMPファイルのロード/セーブなどはできないようですので，なにかの役に立つかもしれません（セーブはできませんが，ロードだけなら，SX-sliceで描画→リセット→MATIERを起動という手順で読むには読めます)。最近は640×480ドットにとどまらず，1280×1024ドットなどという画像データも出回っているようですから，このプログラムを参考に，MATIERの仮想画面にロード/セーブするプログラムを作成するのも面白いかもしれません。

また，MATIERのように多機能なグラフィックツールになると，線を引く・円を描く，などに留まらず，指定範囲をぼかす・メッシュ変形をかけるなどといった機能を使えます。こうなると一から絵を描くだけではなく，既存のデータをつなぎあわせるという方法が使えます。

ランレングスについて

今回はサポートしませんでしたが，BMPファイルのなかにはランレングス(Run-Length)法を用いて圧縮したものも存在します。

たとえば，

00 00 00 00 00 00 01 01 01 01

で示されるような画像データがあった場合，

6個の00

と，

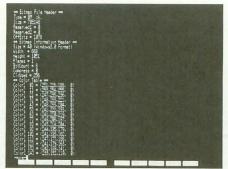
4個の01

という情報に変換したほうがメモリ使用量が少なくて有利です。このような，

00 00 00 00 00 00 01 01 01 01

↓

06 00 04 01

というような変換をランレングス圧縮といいます。

ヘッダ解析の様子

実際のBMPファイルでは，compression＝1だった場合，まさにこのような圧縮方法が採用されています。

このような圧縮方法は同じデータが連続して再現される場合にきわめて有効です。BMPファイルは画像を扱うファイルですので，色変化の少ない絵（アニメ調，シルクスクリーン調のもの）に対してこの圧縮法は有効です。

ただし，

01 02 03 04 05 06

のように変化の激しいデータ列は，

01 01 01 02 01 03 01 04 01 05 01 06

とランレングス圧縮をかけても，データ量を増やすだけですので意味がありません。

また，普通に使っている分には奇数番目の値（偶数番目の値がいくつ続くかを示している数値）は0になりませんから，ここの部分も無駄です。

ですから奇数番目の値＝0のときに特別な意味あいを持たせ，ランレングス圧縮をかけたり，かけないようにしたりして，BMPファイルが無駄に大きくならないようにします。

0，0とデータが続いていたら，そのラインは終了。

0，1とデータが続いていたら，画像データは終了。

0，2とデータが続いていたら，その次に，X，Yを表す2バイトデータが続いているので，走査点を右にXドット，上にYドットずらす。

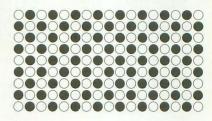
0の次に，3～255の値がきたら，その値の分だけ，ランレングス圧縮をかけないデータが続く。ただし，データが奇数個のときは，1バイト意味のないデータが続く。

例）00 03 45 56 67 00

→ 45 56 67

以上では，データを1バイト＝1ドットで扱っていましたが，compression＝2のBMPファイルは，1バイト＝2ドットで扱います。上位ビット，下位ビットの順に必要な長さ分，繰り返し展開します。

例）06 12

→ 01 02 01 02 01 02

この方法は，

のようなタイリング（このページは基本的に白黒ですが，少し目を離してみると，この配色は灰色に見えてきます。このように少ない色数で，見せかけの色数を増やす方法です。ちなみに，リストが掲載されているところも，虫眼鏡で見れば，上のような模様の集まりであるのがわかると思います）に対して，有効だと思います。

ただ，この圧縮は4ビットを超えるデータを圧縮できません。つまり，16色以下の絵に対してのみ有効となります。この形式のBMPファイルは実際にほとんど存在しないわけで，SX-sliceではサポートしていません。

謝辞

SX-sliceのプログラムは，以下のツールを用いて作成しました。制作者の方々に感謝します。

gcc ver. 1.00 Tool#1 Based on 1.42

HAS ver.2.25

HLK ver.2.27

また，SX-WINDOWの背景を透明にする方法に関しては，grroot.x(箕浦真氏作成)を参考にしました。

参考文献

1） James D.Murray & William van Ryper,Encyclopedia of Graphic file formats,O'Reilly & Associates inc.

2） Microsoft Windows Multimedia Programmer's Reference,MicroSoft Press

リスト1

```
 1: # SX-Slice作成用のメイクファイル
 2:
 3: EXE      = sxslice.x
 4: C_SW        = -O -c -fomit-frame-pointer'-fstrength-reduce -Wall
 5: C_OBJS   = ssmain.o picslice.o bmpslice.o
 6: C_LIBS   = sxlib.l clib.l doslib.l iocslib.l gnulib.l floatfnc.l
 7: C_H         = sxslice.h
 8:
 9: %.o::    %.c $(C_H)
10:     gcc $(C_SW) $<
11:
12: $(EXE): $(C_OBJS)
13:     hlk -l -O $(EXE) $^ $(C_LIBS)
```

リスト2

```
 1: /*********************************************************
 2:  * sxslice.h:              SX-Slice 用ヘッダファイル
 3:  *********************************************************
 4:  *
 5:  * 定数定義
 6:  */
 7: /* ウィンドウタイトル */
 8: #define WINTITLE    ((_LASCII) "¥010SX-Slice")
 9:
10: /* ウィンドウサイズ */
11: #define WIN_H               200             /* ウィンドウ幅       */
12: #define WIN_V               30              /* ウィンドウ高さ     */
13:
14: #define TSIZE_H             80              /* テキストの表示行数   */
15: #define TSIZE_V             10
16: #define LHEIGHT             12              /* テキストの改行幅(ドット) */
17: #define F_READ              0
18:
19: /* イベントマスク */
20: #define EVENTMASK  (EM_MSLDOWN | EM_KEYDOWN | EM_UPDATE | EM_ACTIVATE | EM_SYSTEM1 | EM_SYS
TEM2 | 1)
21:
22: /* 属性マスク */
23: #define ATTRMASK    (TS_SYSTEM | TS_VOLID | TS_SUBDIR | TS_ARCH)
24:
25: /* メッセージの送信に用いるIDナンバー              */
26: /* 0 ～ 127までと負数はシステムが使用する */
27: #define    SX_BASIC_SEND   258
28:
29: #define    TRUE    1
30: #define    FALSE   0
31:
32: /*
33:  * 関数プロトタイプ
34:  */
35: /* ssmain.c */
36: BOOLEAN init(void );
37: BOOLEAN createWindow(void );
38: void msLDownEvent(void );
39: void idleEvent(void );
40: BOOLEAN loadPICfile(char *);
41: void ScreenClr(void );
42: void drawPIC(void );
43: void keyDownEvent(void );
44: void updateEvent(void );
45: void drawGraph(void );
46: void activateEvent(void );
47: void systemEvent(void );
48: void SaveSize(void );
49: void RecovSize(void );
50: void dropIcon(void );
51: void showErrDialog(void );
52: void endProc(int);
53: BOOLEAN loadFile(char *);
54: void SendMes(int, char *, ...);
55:
56: /* bmpslice.c */
57: int         loadBMPfile(char *);
58: int         GetHeaderData(void );
59: void        ColorTableGet(int);
60: void        drawBMP(void );
61: void        DrawBMPScale(void );
62: void        DrawBMPNonScale(void );
63: void        DrawBMPline(void );
64: void        EndOfDraw(void );
65: int         GetColor(void );
66: int         Get2Bytes(void );
67: int         Get4Bytes(void );
```

リスト3

```
  1: #include   <stdio.h>
  2: #include   <string.h>
  3: #include   <doslib.h>
  4: #include   <iocslib.h>
  5: #include   <sxmemory.h>
  6: #include   <sxgraph.h>
  7: #include   <task.h>
  8:
  9: #include   "picslice.h"    /* picslice固有のヘッダファイル */
 10: #include   "sxslice.h"             /* このプログラム固有のヘッダファイル   */
 11:
 12: int                *colorPal;      /* Color pallete table */
 13: int                ioffset;        /* Offset to bitmap data */
 14: int                hsize;          /* Header Size */
 15: int                bitcount;       /* Color bits per Pixel */
 16: int                clrUsed;        /* Number of Colors in Color Table */
 17: int                compress;       /* Compression scheme (0 = none) */
 18: UWORD      *adrs;
 19: int                originY;
 20: int                leftBits;       /* Left Bits for bitcount = 1 or 4 */
 21: int                leftData;
 22: BOOLEAN    scaleFlag;
 23:
 24: extern     BOOLEAN drawing;        /* 描画中か               */
 25: extern     int     y;              /* 描画中のY座標      */
 26: extern     int     dy;             /* 一回の描画幅(縦)   */
 27: extern     picsliceinfo    psi;
 28: extern     char    filename[90];   /* ファイル名          */
 29: Rect       rcDest;                 /* Destination Rectangle */
 30:
 31: int
 32: loadBMPfile(char *fname)
 33: {
 34:    int             fn, r;
 35:
 36:    if(drawing) {
 37:            MMPtrDispose( psi.filebuf );
 38:            psi.filebuf = NULL;
 39:            MMPtrDispose( psi.chainbuf );
 40:            psi.chainbuf = NULL;
 41:    }
 42:    if(psi.filehandle)
 43:            TSClose(psi.filehandle);
 44:
 45:    /* 新たにファイルを開く */
 46:    fn = TSOpen(fname, 0);
 47:    if(fn < 0 ) {
 48:            return PICSLICE_EREAD;
 49:    }
 50:
 51:    /* あらかじめ設定しておくpicsliceinfoのメンバ */
 52:    psi.filehandle = 0;
 53:    psi.filebufsize = SEEK(fn, 0, SEEK_END);
 54:    SEEK(fn, 0, SEEK_SET);
 55:    psi.currentbyte = 0;
 56:    psi.filebuf    = (unsigned char *)MMChPtrNew( psi.filebufsize );
 57:    psi.chainbuf   = NULL;
 58:    leftBits = 0;
 59:
 60:    READ(fn ,psi.filebuf, psi.filebufsize);
 61:
 62:    TSClose(fn);
 63:    r = GetHeaderData();
 64:    if(r != PICSLICE_NORMAL)        return(r);
 65:
 66:    if(clrUsed)     ColorTableGet(hsize + 14);
 67:    else            colorPal = NULL;
 68:    if(compress)    return(PICSLICE_EFORMAT);
 69:
 70: #if 0
 71: {
 72:    char    buff[100];
 73:    sprintf(buff, "(%d,%d)", psi.width, psi.height);
 74:    DMError(1, buff);
 75: }
 76: #endif
 77:    rcDest.l.l_t = 0;
 78:    rcDest.d.right  = psi.width;
 79:    rcDest.d.bottom = psi.height;
 80:    originY = psi.height;
 81:
 82:    if(psi.width <= 512 && psi.height <= 512) {
 83:            scaleFlag = FALSE;
 84:    } else if(psi.width > psi.height) {
 85:            scaleFlag = TRUE;
 86:            rcDest.d.right  = 512;
 87:            rcDest.d.bottom = psi.height * 512 / psi.width;
 88:    } else {
 89:            scaleFlag = TRUE;
 90:            rcDest.d.right  = psi.width * 512 / psi.height;
 91:            rcDest.d.bottom = 512;
 92:    }
 93:
 94:    y = rcDest.d.bottom;
 95:    psi.currentbyte = ioffset;
 96:    strcpy(filename, fname);
 97:    drawing = TRUE;
 98:    drawGraph();
 99:    adrs = (UWORD *)(0xc00000 + 1024 * (y - 1));
100:    return PICSLICE_NORMAL;                         /* 成功 */
101: }
102:
103: int
104: GetHenderData(void )
105: {
106:    int             id;
107:    int             rsv;
108:    int     planes;
109:
110:    id = psi.filebuf[psi.currentbyte++];
111:    id = id*256+ psi.filebuf[psi.currentbyte++];
112:    if ( id != 0x424D ) {           /* Suppurt Only 'BM' format */
113:            return(PICSLICE_EFORMAT);
114:    }
115: #if 0
116:    if(Get4Bytes() != psi.filebufsize)
117:            return(PICSLICE_EFORMAT);
118: #else
119:    Get4Bytes();
120: #endif
121:    rsv= Get4Bytes();
122:    if(rsv) return(PICSLICE_EFORMAT);/* Must be always 0 (Reserved field) */
123:
124:    /* Information area offset */
125:    ioffset  = Get4Bytes();
126:
127:    /* Header size */
128:    hsize= Get4Bytes();
129:
130:    if(hsize==12) {
131: /*         printf("このファイルはOS/2 1.xのフォーマットです¥n");   */
132:    } else if (hsize == 40) {
133: /*         printf("このファイルはWindows3.0のフォーマットです¥n"); */
134:    } else if (hsize == 64) {
135: /*         printf("このファイルはOS/2 2.0のフォーマットです¥n");   */
136:    } else {
137:            return(PICSLICE_EFORMAT);
138:    }
139:
140:    /* Image Size (width, height) In pixel */
141:    if(hsize == 40) {
142:            /* Windows 3.0 */
143:            psi.width  = Get4Bytes();
144:            psi.height = Get4Bytes();
145:    } else {
146:            psi.width  = Get2Bytes();
147:            psi.height = Get2Bytes();
148:    }
149:
150:    planes = Get2Bytes();
151:    if(planes != 1) return(PICSLICE_EFORMAT);       /* Support only single plane */
152:
153:    /* Bit counts */
154:    bitcount  = Get2Bytes();
155:
156:    if(hsize==40 | hsize==64) {
157:            /* Windows 3.0 or OS/2 2.x*/
158:            compress  = Get4Bytes();
159:            clrUsed   = (ioffset - hsize - 14) / 4;
160:    } else {
161:            compress = 0;
162:            clrUsed  = 0;
163:    }
164:    return(PICSLICE_NORMAL);
165: }
166:
167: void
168: ColorTableGet(int toffset)
169: {
170:    int     i;
171:
172:    psi.currentbyte = toffset;
173:
174:    colorPal = (int *)MMChPtrNew(sizeof(int) * clrUsed);
175:    if (hsize==40 || hsize==64 ) {
176:            for (i=0; i< clrUsed; i++){
177:                    short   r, g, b;
178:                    b = psi.filebuf[psi.currentbyte++] / 8;
179:                    g = psi.filebuf[psi.currentbyte++] / 8;
180:                    r = psi.filebuf[psi.currentbyte++] / 8;
181:                    psi.currentbyte++;
182:                    colorPal[i] = (g * 0x400 + r * 0x20 + b) * 2;
183:            }
184:    } else {
185:            /* OS/2 1.x*/
186:            for (i=0;i< clrUsed; i++){
187:                    short   r, g, b;
188:                    b = psi.filebuf[psi.currentbyte++] / 8;
189:                    g = psi.filebuf[psi.currentbyte++] / 8;
190:                    r = psi.filebuf[psi.currentbyte++] / 8;
```

▶18日の今日，フィーバーで儲けたので，Oh!Xがいっぱい買える!?

北浦　真一(25)大阪府

```
191:                    colorPal[i] = (g * 0x400 + r * 0x20 + b) * 2;
192:            }
193:    }
194: }
195:
196: void
197: drawBMP(void )
198: {
199:    if(scaleFlag)   DrawBMPScale();
200:    else                    DrawBMPNonScale();
201: }
202:
203: void
204: DrawBMPScale(void )
205: {
206:    int             sp, i;
207:
208:    sp = SUPER(0);
209:    for(i = 0; i < dy; i++) {
210:            y--;
211:            if((int )adrs < 0xc00000) {
212:                    EndOfDraw();
213:                    break;
214:            }
215:            if(y * psi.height / rcDest.d.bottom < originY) {
216:                    while(0xc00000 <=(int ) adrs
217:                    && y * psi.height / rcDest.d.bottom < originY) {
218:                            originY--;
219:                            DrawBMPline();
220:                    }
221:            } else {
222:                    memcpy(adrs, &adrs[512], 1024);
223:            }
224:            adrs -= 512;
225:    }
226:    SUPER(sp);
227: }
228:
229: void
230: DrawBMPline(void )
231: {
232:    int     nowX, originX = 0;
233:    short   c = 0;
234:
235:    for(nowX = 1; nowX <= rcDest.d.right; nowX++) {
236:            while(originX < nowX * psi.width / rcDest.d.right) {
237:                    if(++originX < psi.width)       c = GetColor();
238:            }
239:            adrs[nowX] = c;
240:    }
241: /* while(++originX < psi.width) GetColor();        */
242:    while(++originX < ((psi.width+3) & (~3))) GetColor();
243:
244:    while((psi.currentbyte - ioffset) & 3)  psi.currentbyte++;
245:    leftBits = 0;
246: }
247:
248: void
249: DrawBMPNonScale(void )
250: {
251:    int             sp, nowX;
252:    short   i;
253:
254:    sp = SUPER(0);
255:    for(i = 0; i < dy; i++) {
256:            y--;
257:            if((int )adrs < 0xc00000) {
258:                    EndOfDraw();
259:                    break;
260:            }
261:            for(nowX = 0; nowX < psi.width; nowX++) {
262:                    short   c;
263:                    c = GetColor();
264:                    if(nowX < 512)          adrs[nowX] = c;
265:            }
266:            while((psi.currentbyte - ioffset) & 3)  psi.currentbyte++;
267:            leftBits = 0;
268:            adrs -= 512;
269:    }
270:    SUPER(sp);
271: }
272:
273: void
274: EndOfDraw(void )
275: {
276:    /* 描画終了 */
277:    MMPtrDispose((Pointer )psi.filebuf );
278:    psi.filebuf = NULL;
279:    MMPtrDispose((Pointer )colorPal);
280:    colorPal = NULL;
281:    drawing = FALSE;
282:    drawGraph();
283: }
284:
285: int
286: GetColor(void )
287: {
288:    short   c;
289:
290:    switch(bitcount) {
291:            case 1:
292:                    if(--leftBits <= 0) {
293:                            leftData = psi.filebuf[psi.currentbyte++];
294:                            leftBits = 8;
295:                    }
296:                    if(leftData & 0x80)     c = 1;
297:                    else                    c = 0;
298:                    leftData <<= 1;
299:                    break;
300:            case 4:
301:                    if(--leftBits <= 0) {
302:                            leftData = psi.filebuf[psi.currentbyte++];
303:                            leftBits = 2;
304:                    }
305:                    c = (leftData >> 4) & 0x0f;
306:                    leftData <<= 4;
307:                    break;
308:            case 8:
309:                    c = psi.filebuf[psi.currentbyte++];
310:                    break;
311:            case 24:
312:            default:
313:                    c = 0;
314:                    break;
315:    }
316:
317:    if(clrUsed) {
318:            return(colorPal[c]);
319:    } else {
320:            short   r, g, b;
321:            b = psi.filebuf[psi.currentbyte++] / (1 << (bitcount/3 - 5));
322:            g = psi.filebuf[psi.currentbyte++] / (1 << (bitcount/3 - 5));
323:            r = psi.filebuf[psi.currentbyte++] / (1 << (bitcount/3 - 5));
324:            return((g * 0x400 + r * 0x20 + b) * 2);
325:    }
326: }
327:
328: int
329: Get2Bytes(void )
330: {
331:    int     ret;
332:
333:    ret = psi.filebuf[psi.currentbyte++];
334:    ret += psi.filebuf[psi.currentbyte++]*0x100;
335:    return(ret);
336: }
337:
338: int
339: Get4Bytes(void )
340: {
341:    int     ret;
342:
343:    ret = psi.filebuf[psi.currentbyte++];
344:    ret += psi.filebuf[psi.currentbyte++]*0x100;
345:    ret += psi.filebuf[psi.currentbyte++]*0x10000;
346:    ret += psi.filebuf[psi.currentbyte++]*0x1000000;
347:    return(ret);
348: }
```

リスト4

```
  1: /****************************************
  2:  * sx.c:   PicSlice For SX-WINDOW
  3:  *         Programmed By ISHIGAMI Tatsuya
  4:  *                                05/06/94
  5:  ****************************************
  6: */
  7:
  8: #include <stdio.h>
  9: #include <fctype.h>
 10: #include <stdlib.h>
 11: #include <string.h>
 12: #include <doslib.h>
 13: #include <sxmemory.h>    /* メモリマンを利用するときに必要        */
 14: #include <event.h>       /* イベントマンを利用するときに必要      */
 15: #include <sxgraph.h>     /* グラフ系マネージャを利用するときに必要 */
 16: #include <window.h>      /* ウィンドウマンを利用するときに必要    */
 17: #include <dialog.h>      /* ダイアログマンを利用するときに必要    */
 18: #include <text.h>        /* テキストマンを利用するときに必要      */
 19: #include <task.h>        /* タスクマンを利用するときに必要        */
 20: #include "picslice.h"    /* picslice固有のヘッダファイル */
 21: #include "sxslice.h"     /* このプログラム固有のヘッダファイル    */
 22:
 23: char _sxkernelcomm[] = SXKERNEL;  /* カーネル起動コマンド          */
 24:
 25: /*
 26:  * グローバル変数
 27:  */
 28: Window *windowPtr;           /* ウィンドウポインタ   */
 29: BOOLEAN activeFlag;          /* アクティブフラグ     */
 30: BOOLEAN drawing;             /* 描画中か           */
 31: TsEvent event;               /* イベントレコード     */
 32: int         eventMask;       /* イベントマスク       */
 33: int         errorCode;       /* エラーコード         */
 34: BOOLEAN endFlag;             /* 終了フラグ           */
 35: char        filename[90];    /* ファイル名           */
 36: int         y;               /* 描画中のY座標        */
 37: int         dy;              /* 一回の描画幅（縦）   */
 38: BOOLEAN     visibleFlag;     /* ウィンドウを表示するか */
 39: picsliceinfo        psi;
 40:
 41: typedef     enum {
 42:     FL_PIC,
 43:     FL_BMP
 44: } FILEKIND;
 45: FILEKIND    filekind;
 46:
 47: /* バッファのサイズ(単位:バイト) */
 48: #define     FILEBUFSIZE     4096
 49: #define     CHAINBUFSIZE    8196
 50:
 51: /* 分割ロードの一回あたりのラスタ数 */
 52: #define     RASTER  8
 53:
 54:
 55: /**************************************************************************
 56:  * main():         メイン関数
 57:  **************************************************************************
 58:  */
 59: int
 60: main(void)
 61: {
 62:
 63:    if (!init()) {           /* 初期化処理を行う                  */
 64:            showErrDialog();         /* エラーダイアログを表示する    */
 65:            endFlag = TRUE; /* 終了フラグをセットする        */
 66:    } else {
 67:            short   taskID = 0;     /* タスクID */
 68:            do {
 69:                    taskID = TSFindTskn("sxslice.x", taskID);
 70:            } while(taskID == TSGetID());
 71:            if(taskID > 0) { /* すでに起動されている場合 */
 72:                    if(filename[0]) SendMes(taskID, "FILE %s", filename);
 73:                    endFlag = TRUE; /* 終了フラグをセットする           */
 74:            } else {
 75:                    if(filename[0]) loadFile(filename);
 76:                    WMShow(windowPtr);
 77:                    visibleFlag = TRUE;
 78:            }
 79:    }
 80:    if(!endFlag && visibleFlag) WMShow(windowPtr);
 81:
 82:    /* 終了フラグがセットされるまでループする */
 83:    while (!endFlag) {
 84:            errorCode = 0;  /* エラーコードをクリアする      */
 85:            /* タスクマンからイベントを取得する */
 86:            TSEventAvail(eventMask, &event);
 87:            /* 自分をカレントグラフにする */
 88:            GMSetGraph(&windowPtr->graph);
 89:            /* 各イベントに対応した処理を行う */
 90:            switch (event.ts.what) {
 91:            case E_MSLDOWN:         /* マウス左ボタンダウンイベント */
 92:                    msLDownEvent();
 93:                    break;
 94:            case E_KEYDOWN:         /* キーダウンイベント           */
 95:                    keyDownEvent();
 96:                    break;
 97:            case E_IDLE:            /* アイドルイベント             */
 98:                    idleEvent();
 99:                    break;
100:            case E_UPDATE:          /* アップデートイベント         */
101:                    updateEvent();
102:                    break;
103:            case E_ACTIVATE:        /* アクティベートイベント       */
104:                    activateEvent();
105:                    break;
106:            case E_SYSTEM1:         /* システムイベント             */
```

▶風邪をひきました。「バカではなかったんだ」と，ホッとしました。

吉行　忠義(26)静岡県

```
107:             case E_SYSTEM2:
108:                     systemEvent();
109:                     break;
110:             }
111:             if (errorCode != 0)     /* エラーが発生したか？          */
112:                     showErrDialog(); /* エラーダイアログを表示する */
113:     }
114:     /* 終了手続きを行う */
115:     endProc((errorCode == 0) ? EXIT_SUCCESS : EXIT_FAILURE);
116:
117:     return 0;
118: }
119:
120: /***************************************************************************
121:  * init():          初期化処理
122:  ***************************************************************************
123:  * 戻り値：BOOLEAN          = TRUE:  初期化成功
124:  *                          = FALSE: 初期化失敗（終了）
125:  * 注釈：  共通変数の初期化、アボート処理関数の設定、ウィンドウの作成と
126:  *         表示などを行う。
127:  */
128: BOOLEAN
129: init(void)
130: {
131:     windowPtr = NULL;       /* ウィンドウポインタ          */
132:     activeFlag = FALSE;     /* アクティブフラグ            */
133:     drawing = FALSE;
134:     eventMask = EVENTMASK;  /* イベントマスク              */
135:     errorCode = 0;          /* エラーコード                */
136:     endFlag = FALSE;        /* 終了フラグ                  */
137:     dy = RASTER;
138:     visibleFlag = FALSE;
139:
140:     TSSetAbort(endProc, NULL); /* アボート処理関数を設定する      */
141:
142:     if (LOWWORD(SXVer()) < SXVER) { /* ＳＸシステムのバージョンを取得する */
143:             errorCode = 1;  /* バージョンが古い            */
144:             return FALSE;   /* 失敗したのでFALSEを返す     */
145:     }
146:
147:     if (!createWindow()) {  /* ウィンドウを作成する        */
148:             errorCode = 2;  /* 作成できなかった            */
149:             return FALSE;   /* 失敗したのでFALSEを返す     */
150:     }
151:
152:     RecovSize();
153:     return TRUE;            /* 成功したのでTRUEを返す      */
154: }
155:
156: /***************************************************************************
157:  * createWindow(): ウィンドウの作成
158:  ***************************************************************************
159:  * 戻り値：BOOLEAN          = TRUE:  作成成功
160:  *                          = FALSE: 作成失敗（終了）
161:  */
162: BOOLEAN
163: createWindow(void )
164: {
165:     int paramFlags;                 /* パラメータの有無            */
166:     Rect rc;
167:     Task task;                      /* タスク管理テーブル          */
168:
169:     static Rect winSize = { 0, 0, WIN_H, WIN_V }; /* ウィンドウサイズ */
170:
171:     TSGetTdb(&task, TS_OWN);        /* タスク管理テーブルを取得する */
172:     paramFlags = TSTakeParam(       /* コマンドラインを解析する     */
173:             task.command,           /* コマンドライン               */
174:             &rc,                    /* 指定されたウィンドウ位置     */
175:             NULL,                   /* 最終文字列を格納しない       */
176:             0,                      /* アドレステーブルを保存しない */
177:             NULL,                   /* アドレステーブルを作成しない */
178:             NULL);                  /* ポインタを確保しない         */
179:     if (!(paramFlags & 1)) {        /* ウィンドウ位置の指定があるか？ */
180:             /* ない場合、システムからウィンドウの左上座標を取得する */
181:             rc = winSize;
182:             GMSlideRect(&rc, TSGetWindowPos());
183:     }
184:     windowPtr = WMOpen(     /* ウィンドウをオープンする     */
185:             NULL,                   /* ウィンドウポインタを確保する */
186:             &rc,                    /* ウィンドウの表示位置         */
187:             WINTITLE,               /* ウィンドウタイトル           */
188:             FALSE,                  /* オープン時に描画しない       */
189:             WI_STD2 << 4,           /* タイトルの広い標準ウィンドウ */
190:             WM_FOREMOST,            /* 一番前に表示する             */
191:             TRUE,                   /* クローズボタンを使用する     */
192:             TSGetID());             /* 自分のタスクＩＤ             */
193:
194:     if (windowPtr == NULL)
195:             return FALSE;           /* 失敗したのでFALSEを返す      */
196:
197:     return TRUE;                    /* 成功したのでTRUEを返す       */
198: }
199:
200: /***************************************************************************
201:  * msLDownEvent(): マウス左ボタンダウンイベント処理
202:  ***************************************************************************
203:  * 注釈：
204:  * ウィンドウ上でマウスの左ボタンが押された場合の処理を行う。
205:  * この処理でクローズボタンによる終了、ウィンドウの移動が可能となります。
206:  */
207: void
208: msLDownEvent(void )
209: {
210:     int partCode;                   /* ウィンドウのパートコード     */
211:     Window *wTemp;                  /* テンポラリウィンドウポインタ */
212:
213:     /* イベントが自分のウィンドウか？ */
214:     if (event.ev.whom.win == windowPtr) {
215:             /* ウィンドウがインアクティブで、OPT.1キーが押されてないか？ */
216:             if (!activeFlag && !(event.ev.how & (KS_OPT1 | KS_XF1))) {
217:                     /* ウィンドウをアクティブにする */
218:                     WMSelect(windowPtr);
219:                     /* ボタンが押された場所のパートコードを取得する */
220:                     partCode = WMFind(event.ev.where.x_y, &wTemp);
221:                     /* タイトルバー以外か、左ボタンが離されたか？ */
222:                     if (partCode != W_INDRAG || !EMLStill())
223:                             return;
224:             }
225:             /* マウスのボタンが押されている間、ウィンドウの各種処理をシス
226:              * テムに任せて、ボタンが離された場所のパートコードを取得する
227:              */
228:             partCode = SXCallWindM(windowPtr, &event);
229:             if (partCode == W_INCLOSE) /* クローズボタンか？          */
230:                     endFlag = TRUE; /* 終了フラグをセットする      */
231:     }
232: }
233:
234: void
235: idleEvent(void )
236: {
237:     if(drawing == FALSE)    return;
238:
239:     if(filekind == FL_PIC) {
240:             drawPIC();
241:     } else if(filekind == FL_BMP) {
242:             drawBMP();
243:     }
244: }
245:
246: void
247: drawPIC(void)
248: {
249:     int             sp, ret;
250:
251:     /* スーパバイザモード(G-RAMに直接書き込むため) */
252:     sp = SUPER( 0 );
253:
254:     /* ローディングのメインループ */
255:     ret = picsliceLoad( &psi, (unsigned short*)(0xC00000+y*1024), dy );
256:     y += dy;
257:     switch ( ret ) {
258:     case PICSLICE_NORMAL:
259:             break;
260:     case PICSLICE_TERMINATE:
261:     case PICSLICE_EREAD:
262:     case PICSLICE_EOVERFLOW:
263:             /* 後始末 */
264:             MMPtrDispose( psi.filebuf );
265:             psi.filebuf = NULL;
266:             MMPtrDispose( psi.chainbuf );
267:             psi.chainbuf = NULL;
268:             if(psi.filehandle ) {
269:                     TSClose(psi.filehandle);
270:                     psi.filehandle = 0;
271:             }
272:             drawing = FALSE;
273:             drawGraph();
274:             break;
275:     }
276:     /* ユーザモード */
277:     SUPER( sp );
278: }
279:
280: /***************************************************************************
281:  * keyDownEvent(): キーダウンイベント処理
282:  ***************************************************************************
283:  * 注釈：  OPT.1+'Q'キーでの終了処理を行う。
284:  */
285: void
286: keyDownEvent(void )
287: {
288:     int shortCut;                   /* ショートカットキー          */
289:
290:     if (event.ev.how & (KS_OPT1 | KS_XF1)) { /* OPT.1キーが押されたか？      */
291:             /* キー入力した文字を大文字に変換する */
292:             shortCut = toupper((int) event.ev.whom.key.ascii);
293:             if (shortCut == 'Q')    /* 終了か？                    */
294:                     endFlag = TRUE; /* 終了フラグをセットする      */
295:     }
296: }
297:
298: /***************************************************************************
299:  * updateEvent():  アップデート処理
300:  ***************************************************************************
301:  * 注釈：
302:  * アップデート処理としてウィンドウ内部を描画する。
303:  * この処理を行わないと、自分より下のウィンドウのアップデートやアイドル
304:  * イベントの処理ができなくなります。
305:  */
306: void
307: updateEvent(void )
308: {
309:     /* イベントが自分のウィンドウか？ */
310:     if (event.ev.whom.win == windowPtr) {
311:             WMUpdate(windowPtr); /* アップデートを開始する */
312:             drawGraph();          /* ウィンドウ内部を描画する    */
313:             WMUpdtOver(windowPtr); /* アップデートを終了する     */
314:     }
315: }
316:
317: /***************************************************************************
318:  * drawGraph():    ウィンドウ内部の描画
319:  ***************************************************************************
320:  */
321: void
322: drawGraph(void )
323: {
324:     GMSetGraph(&windowPtr->graph);
325:
326:     /* ペンモードをバックカラーにする */
327:     GMPenMode(G_BACK << 8 | G_PSET);
328:     GMFillRect(&windowPtr->graph.rect); /* ウィンドウ内をクリアする */
329:     if(drawing) {
330:             char    buff[100];
331:             GMMove(0x00080008);
332:             sprintf(buff, "%sを描画中", filename);
333:             GMDrawStrZ(buff);
334:     }
335: }
336:
337: /***************************************************************************
338:  * activateEvent():         アクティベートイベント処理
339:  ***************************************************************************
340:  * 注釈：  アクティブ、インアクティブによるアクティブフラグの切り替えと、
341:  *         イベントマスクの切り替えを行う。
342:  */
343: void
344: activateEvent(void )
345: {
346:     /* イベントが自分のウィンドウか？ */
347:     if (event.ev.whom.win == windowPtr) {
348:             activeFlag = TRUE;      /* アクティブフラグをセットする */
349:             /* アクティブ時のイベントマスクをセットする */
350:             eventMask |= EM_KEYDOWN;
351:     /* イベントが他のウィンドウで、自分がアクティブ状態か？ */
352:     } else if (activeFlag) {
353:             activeFlag = FALSE; /* アクティブフラグをクリアする */
354:             /* インアクティブ時のイベントマスクをセットする */
355:             eventMask &= (~EM_KEYDOWN);
356:     }
357: }
358:
359: /***************************************************************************
360:  * systemEvent():  システムイベント処理
361:  ***************************************************************************
362:  * 注釈：  全ウィンドウのクローズ、全タスクの終了、ウィンドウのセレクトに
363:  *         対応した処理を行う。
364:  */
365: void
366: systemEvent(void )
367: {
368:     char    *p;
369:
370:     switch (event.ts.what2) {       /* イベントの種類は？          */
371:     case CLOSEALL:                  /* 全ウィンドウのクローズ      */
372:     case ENDTSK:                    /* 全タスクの終了              */
373:             endFlag = TRUE;         /* 終了フラグをセットする      */
374:             break;
375:     case WINDOWSELECT:              /* ウィンドウのセレクト        */
376:             WMSelect(windowPtr); /* ウィンドウをアクティブにする */
377:             break;
378:     case DRAGEND:                   /* ドラッグ終了                */
379:             /* イベントが自分のウィンドウか？ */
380:             if (event.ev.whom.win == windowPtr)
381:                     dropIcon();     /* アイコンのドロップ処理を行う */
382:             break;
```

▶受験生にとって欲しいゲームがないということはたいへんよいのだが，最近のX68000の新作の本数には悲しいものがある。　小野寺　健一(17)宮城県

```
383:    case SAVE:
384:            SaveSize();
385:            break;
386:    case SX_BASIC_SEND:
387:            MMHdlLock((Handle )event.ts.whom);
388:            p = *(char **)(event.ts.whom);
389:            if(!strcmp("SHOW", p) && visibleFlag == FALSE) {
390:                    WMShow(windowPtr);
391:                    visibleFlag = TRUE;
392:            } else if(!strcmp("HIDE", p) && visibleFlag == TRUE) {
393:                    WMHide(windowPtr);
394:                    visibleFlag = FALSE;
395:            } else if(!strncmp("FILE", p, 4)) {
396:                    loadFile(&p[5]);
397:            } else if(!strncmp("RASTER", p, 6)) {
398:                    sscanf(&p[7], "%d", &dy);
399:            } else if(!strncmp("QUIT", p, 4) || !strncmp("END", p, 3)) {
400:                    endFlag = TRUE; /* 終了フラグをセットする      */
401:            }
402:            MMHdlUnlock((Handle )event.ts.whom);
403:            break;
404:    }
405: }
406:
407:
408: /*********************************************************
409: * SaveSize() コマンドラインにウィンドウのサイズと        *
410: *                ドラッグされたファイル名(1つ)を書き込む *
411: *
           *
412: * ウィンドウの位置はその後にシステムが書き込む           *
413: *********************************************************/
414: void
415: SaveSize(void )
416: {
417:    Task    taskBuf;
418:
419:    TSGetTdb(&taskBuf, -1);
420:    sprintf((char *)&taskBuf.command[1], "-R%d -V%d -F%s",
421:            dy, visibleFlag, filename);
422:    taskBuf.command[0] = strlen((char *)&taskBuf.command[1]);
423:    TSSetTdb(&taskBuf, -1);
424: }
425:
426: /*********************************************************
427: * RecovSize() コマンドラインからファイル名を取り出す
428: *********************************************************/
429: void
430: RecovSize(void )
431: {
432:    Task    taskBuf;
433:    char    *p, *next;
434:
435:    filename[0] = 0;
436:
437:    TSGetTdb(&taskBuf, -1);
438:    next = (char *)&taskBuf.command[1];
439:    while(next != NULL) {
440:            p = next;
441:            next = strchr(p ,' ');
442:            if(next != NULL) *next++ = '¥0';
443:            if(*p == '-' || *p == '/') {
444:                    p++;
445:                    switch(toupper(*p++)) {
446:                            case 'F':
447:                                    strcpy(filename, p);
448:                                    break;
449:                            case 'R':
450:                                    sscanf(p, "%d", &dy);
451:                                    break;
452:                            case 'V':
453:                                    sscanf(p, "%d", &visibleFlag);
454:                                    break;
455:                    }
456:            }
457:    }
458: }
459:
460:
461: /**************************************************************************
462:  * dropIcon():     アイコンのドロップ処理
463:  **************************************************************************
464:  */
465: void
466: dropIcon(void )
467: {
468:    int errCode, len;
469:    Rect rc;
470:    Drag *dragPtr;                  /* ドラッグポインタ             */
471:    Cell *pcell;                    /* セルレコードへのポインタ     */
472:    IcState *pis;                   /* アイコン管理レコードへのポインタ */
473:    char fileName[TS_NAMEMAX];      /* ファイル名                   */
474:
475:    /* ドラッグポインタを取得する */
476:    errCode = TSGetDrag(&dragPtr);
477:    if (errCode < 0)
478:            /* ドラッグレコードが無い */
479:            return;
480:
481:    /* セルリストハンドルをロックする */
482:    MMHdlLock(dragPtr->cellListHdl);
483:    /* セルレコードへのポインタを取得する */
484:    pcell = *dragPtr->cellListHdl;
485:    /* ラバーバンドを消去する */
486:    TSHideDrag();
487:    /* セルレコードの情報の種類がアイコン管理レコード(上位ワードが'FS')か?
*/
488:    if (HIWORD(pcell->kind) == 'FS') {
489:            /* アイコンは1つ(ドラッグレコードのセルリストの長さがアイコン
490:             * 管理レコードを含むセルレコードの長さと同一の場合)か?
491:             */
492:            if (dragPtr->length == sizeof(IcState) + 8 ) {
493:                    /* アイコン管理レコードへのポインタを取得する */
494:                    pis = (IcState *) pcell->data;
495:                    /* ファイルの属性をチェック */
496:                    if ((pis->attr & ATTRMASK) == TS_ARCH) {
497:                            /* アイコンのフルパスを取得する */
498:                            len = TSISRecToStr(
499:                                    pis, /* アイコン管理レコード    */
500:                                    fileName); /* パス名格納ポインタ */
501:                            if (len != 0) {
502:                                    /* ドラッグを終了する */
503:                                    TSEndDrag(TS_FINISH);
504:                                    /* ドラッグされたファイルの読み込み */
505:                                    loadFile(fileName);
506:                            }
507:                    } else          /* ファイルアイコン以外の場合   */
508:                            /* アイコンを元の位置までドラッグして終了する */
509:                            TSEndDrag(TS_PUTBACK);
510:            } else
511:                    /* アイコン2つ以上か他のドラッグレコードだったら */
512:                    /* アイコンを元の位置までドラッグして終了する */
513:                    TSEndDrag(TS_PUTBACK);
514:    }
515:    /* セルリストハンドルのロックを解除する */
516:    MMHdlUnlock(dragPtr->cellListHdl);
517:    /* ドラッグされたファイルの内容を描画する */
518:    rc = windowPtr->graph.rect; /* ウィンドウ内全体を書き換える */
519:    GMSlideRect(&rc, GMLocalToGlobal(0)); /* グローバル座標系に変換する */
520:    WMAddRect(&rc);                 /* アップデートリージョンに追加する */
521: }
522:
523: /**************************************************************************
524:  * showErrDialog():        エラーダイアログの表示
525:  **************************************************************************
526:  */
527: void
528: showErrDialog(void )
529: {
530:    int i;
531:
532:    static struct {                 /* エラーメッセージ     */
533:            int code;               /* エラーコード         */
534:            int manager;            /* 使用するマネージャ   */
535:            int flag;               /* フラグ情報           */
536:            char *str;              /* 表示する文字列       */
537:    } errorMsg[] = {
538:            {  1, 1, 1, "SX SYSTEMのバージョンが違います。" },
539:            {  2, 2, 1, "ウィンドウが作成できません。" },
540:            { 11, 1, 1, "ファイルがオープンできません。" },
541:            {  0, 1, 1, "エラーが発生しました。" }
542:    };
543:
544:    for (i = 0; errorMsg[i].code != 0; i++) /* エラーコードを捜す   */
545:            if (errorMsg[i].code == errorCode)
546:                    break;
547:    switch (errorMsg[i].manager) {  /* 利用するマネージャの選択     */
548:    case 1:                         /* ダイアログマン               */
549:            DMError(errorMsg[i].flag, errorMsg[i].str);
550:            break;
551:    case 2:                         /* タスクマン                   */
552:            TSErrDialogN(errorMsg[i].flag, errorMsg[i].str);
553:            break;
554:    }
555: }
556:
557: /**************************************************************************
558:  * endProc():      終了手続き
559:  **************************************************************************
560:  * 引数:  int code        終了コード
561:  * 注釈:  ハンドルの解放やウィンドウの廃棄と、プログラムの終了を行う。
562:  */
563: void
564: endProc(int code )
565: {
566:    /* ウィンドウポインタが確保されたままか? */
567:    if (windowPtr != NULL)
568:            WMDispose(windowPtr); /* ウィンドウを廃棄する   */
569:
570:    /* ファイルが開かれたままか? */
571:    if(psi.filehandle) {
572:            TSClose(psi.filehandle);
573:            psi.filehandle = 0;
574:    }
575:
576:    exit(code);                     /* プログラムを終了する         */
577: }
578:
579: int
580: loadFile(char *fname)
581: {
582:    short   id;
583:    int     r;
584:
585:    if(r = loadBMPfile(fname), r == PICSLICE_NORMAL) {
586:            filekind = FL_BMP;
587:    } else if(r = loadPICfile(fname), r == PICSLICE_NORMAL) {
588:            ScreenClr();
589:            filekind = FL_PIC;
590:    } else {
591:            if(r == PICSLICE_EREAD)
592:                    TSErrDialogN(1, "ファイルがオープンできません");
593:            else
594:                    TSErrDialogN(1, "未対応のフォーマットです");
595:            return(FALSE);
596:    }
597:    id = TSFindTskn("壁紙動画.r", -1);
598:    if(id > 0) {
599:            SendMes(id, "STOP");
600:            SendMes(id, "HOME 0,0");
601:    }
602:    return(TRUE);
603: }
604:
605: void
606: ScreenClr(void )
607: {
608:    int     *p, sp;
609:    short   x,y;
610:
611:    sp = SUPER(0);
612:    p = (int *)0xc00000;
613:    for(y = 0; y < psi.height; y++) {
614:            for(x = 0; x < psi.width / 2; x++) {
615:                    p[x] = 0;
616:            }
617:            *(int *)&p = ((int)p & (~0x3ff)) + 0x400;
618:    }
619:    SUPER(sp);
620: }
621:
622: /**************************************************************************
623:  * loadPICfile():  ファイル読み込み
624:  **************************************************************************
625:  *         char *pname     ファイル名へのポインタ
626:  * 戻り値: BOOLEAN          = TRUE:  読み込み成功
627:  *                         = FALSE: 読み込み失敗
628:  */
629: BOOLEAN
630: loadPICfile(char *pname)
631: {
632:    int             fn, r;
633:
634:    /* ファイルが開かれたままか? */
635:    if(drawing) {
636:            MMPtrDispose( psi.filebuf );
637:            psi.filebuf = NULL;
638:            MMPtrDispose( psi.chainbuf );
639:            psi.chainbuf = NULL;
640:    }
641:    if(psi.filehandle) {
642:            TSClose(psi.filehandle);
643:            psi.filehandle = 0;
644:    }
645:
646:    /* 新たにファイルを開く */
647:     if ( (fn = TSOpen( pname, 0 )) < 0 ) {
648:            errorCode = 11; /* オープンできなかった         */
649:            return PICSLICE_EREAD;
650:    }
651:
652:    /* あらかじめ設定しておくpicsliceinfoのメンバ */
653:    psi.filehandle = fn;
654:    psi.filebuf = (unsigned char *)MMChPtrNew( FILEBUFSIZE );
655:    psi.filebufsize = FILEBUFSIZE;
656:    psi.chainbuf = (unsigned short *)MMChPtrNew( CHAINBUFSIZE );
```

▶「複数のX680x0によるフルカラーアニメーションに関する研究」には本当に驚かされました。
千馬　宏(20)熊本県

```
657:    psi.chainbufsize = CHAINBUFSIZE/sizeof(unsigned short);
658:    /* picsliceローディングの開始 */
659:    r = picsliceOpen( &psi, PICSLICE_GRAM );
660:    if(r != PICSLICE_NORMAL) {
661:            TSClose(psi.filehandle);
662:            psi.filehandle = 0;
663:            MMPtrDispose((Pointer)psi.filebuf);
664:            psi.filebuf = NULL;
665:            MMPtrDispose((Pointer)psi.chainbuf);
666:            psi.chainbuf = NULL;
667:            return(r);
668:    }
669:
670:    strcpy(filename, pname);
671:    drawing = TRUE;
672:    drawGraph();
673:    y = 0;
674:    return PICSLICE_NORMAL;
675: }
676:
677: /********************************/
678: /*         文字列を送信する              */
679: /********************************/
680: void
681: SendMes(int id, char *fmt, ...)
682: {
683:    char    **hdl, *p, *q, *sval;
684:    int             i, ret;
685:    double  f;
686:    int     cnt = 10;       /*メッセージのやりとりを10回行なって
687:                                    それでもダメならエラーにする         */
688:    TsEvent eventRec;
689:    va_list ap;     /* 各名なし引数を順々に渡す */
690:
691:    va_start(ap, fmt);
692:    hdl = (char **)MMChHdlNew(1024);
693:    if(hdl == NULL) {
694:            TSErrDialogN(0x101,"メモリが確保出来ません");
695:            return;
696:    }
697:    q = *hdl;
698:    for(p = fmt; *p; p++) {
699:            if(*p != '%') {
700:                    *q++ = *p;
701:                    continue;
702:            }
703:            switch(*++p) {
704:                    case 'd':
705:                            i = va_arg(ap, int);
706:                            ret = sprintf(q, "%d", i);
707:                            q = &q[ret];
708:                            break;
709:                    case 'f':
710:                            f = va_arg(ap, double);
711:                            ret = sprintf(q, "%f", f);
712:                            q = &q[ret];
713:                            break;
714:                    case 's':
715:                            sval = va_arg(ap, char *);
716:                            while(*sval)    *q++ = *sval++;
717:                            break;
718:                    default:
719:                            *q++ = *p;
720:                            break;
721:            }
722:    }
723:    va_end(ap);     /* クリーンアップ */
724:    *q = '¥0';
725:
726:    eventRec.ts.whom   = (long)hdl;
727:    eventRec.ts.when   = EMSysTime();
728:    eventRec.ts.what2  = SX_BASIC_SEND;
729:    do {
730:            ret = TSSendMes(id, &eventRec);
731:            cnt--;
732:    } while(ret != 0 && ret != 2 && cnt != 0);
733:
734:    MMHdlDispose(hdl);
735:    if(cnt == 0) {
736:            TSErrDialogN(1, "無効なタスクに通信を行ないました");
737:    }
738: }
```

リスト5

```
  1: /*
  2: ** BMP File Information print program.
  3: **         Programmed by ISHIGAMI Tatsuya
  4: **                                  '94 Nov 10th
  5: */
  6:
  7: #include   <stdio.h>
  8:
  9: int        ioffset;        /* Offset to bitmap data */
 10: int        hsize;          /* Header Size */
 11: int        clrUsed;        /* Number of Colors in Color Table */
 12: int        iwidth, iheight;/* Size of Bitmap in Pixels */
 13: int        compress;       /* Compression scheme (0 = none) */
 14:
 15: void       Init(void );
 16: void       FileHeader(FILE *);
 17: void       InformationHeader(FILE *);
 18: void       ColorTable(FILE *, int);
 19: int        Get2Bytes(FILE *);
 20: int        Get4Bytes(FILE *);
 21:
 22: void
 23: main(int argc, char *argv[1])
 24: {
 25:    FILE    *fp;
 26:    int             i, j;
 27:
 28:    if(argc != 2) {
 29:            puts("Usage: bmpinfo filename");
 30:            exit(-1);
 31:    }
 32:    fp= fopen(argv[1], "rb");
 33:    if((int )fp <= 0) {
 34:            printf("Can not open %s¥n", argv[1]);
 35:            exit(-1);
 36:    }
 37:    FileHeader(fp);
 38:    InformationHeader(fp);
 39:
 40:    if(clrUsed)     ColorTable(fp, hsize + 14);
 41:
 42:    fclose(fp);
 43: }
 44:
 45: void
 46: FileHeader(FILE *fp)
 47: {
 48:    int     id;
 49:    int     rsv;
 50:    int     psize;
 51:    int     planes;
 52:
 53:    printf("== Bitmap File Header ==¥n");
 54:
 55:    id = fgetc(fp);
 56:    id = id*256+ fgetc(fp);
 57:    if ( id == 0x424D ) {           /* Check 'BM' format */
 58:            printf("Type = BM  ok.¥n");
 59:    } else {
 60:            printf ("Type = %x  incorrect.¥n", id);
 61:            return;
 62:    }
 63:
 64:    printf("Size = %d¥n", Get4Bytes(fp));
 65:    rsv = Get2Bytes(fp);
 66:    printf("Reserved1 = %d¥n", rsv);
 67:    if(rsv)         return;
 68:    rsv = Get2Bytes(fp);
 69:    printf("Reserved2 = %d¥n", rsv);
 70:    if(rsv)         return;
 71:
 72:    /* Information area offset */
 73:    ioffset  = Get4Bytes(fp);
 74:    printf("OffBits = %d¥n", ioffset);
 75:
 76: }
 77:
 78:
 79: void
 80: InformationHeader(FILE *fp)
 81: {
 82:    printf("== Bitmap Information Header ==¥n");
 83:
 84:    /* Header size */
 85:    hsize= Get4Bytes(fp);
 86:    printf("Size = %d ", hsize);
 87:
 88:    switch(hsize) {
 89:            case 12:
 90:                    printf("(OS/2 1.x Format)¥n");
 91:                    break;
 92:            case 40:
 93:                    printf("(Windows3.0 Format)¥n");
 94:                    break;
 95:            case 64:
 96:                    printf("(OS/2 2.0 Format)¥n");
 97:                    break;
 98:            defalut:
 99:                    printf("...Incorrect.¥n");
100:                    return;
101:    }
102:
103:    /* Image Size (iwidth, iheight) In pixel */
104:    if(hsize == 40) {
105:            /* Windows 3.0 only **/
106:            iwidth  = Get4Bytes(fp);
107:            iheight = Get4Bytes(fp);
108:    } else {
109:            iwidth  = Get2Bytes(fp);
110:            iheight = Get2Bytes(fp);
111:    }
112:    printf("Width  = %d¥n", iwidth);
113:    printf("Height = %d¥n", iheight);
114:    printf("Planes = %d¥n", Get2Bytes(fp));
115:    printf("BitCount = %d¥n", Get2Bytes(fp));
116:
117:    if(hsize==40 | hsize==64) {
118:            /* Windows 3.0 or OS/2 2.x*/
119:            compress  = Get4Bytes(fp);
120: #if 0
121:            fseek(fp, 12, SEEK_CUR);
122:            clrUsed   = Get4Bytes(fp);
123: #else
124:            clrUsed   = (ioffset - hsize - 14) / 4;
125: #endif
126:    } else {
127:            compress = 0;
128:            clrUsed  = 0;
129:    }
130:    printf("Compress = %d¥n", compress);
131:    printf("ClrUsed = %d¥n", clrUsed);
132: }
133:
134: void
135: ColorTable(FILE *fp, int toffset)
136: {
137:    int     i;
138:
139:    printf("== Color Table ==¥n");
140:
141:    fseek(fp, toffset, SEEK_SET);
142:    if (hsize==40 || hsize==64 ) {
143:            for (i=0; i< clrUsed; i++){
144:                    short   r, g, b, rsv;
145:                    b = fgetc(fp);
146:                    g = fgetc(fp);
147:                    r = fgetc(fp);
148:                    rsv = fgetc(fp);
149:                    printf("Color[%3d] = (%3d,%3d,%3d,%3d)¥n", i, r, g, b, rsv);
150:            }
151:    } else {
152:            /* OS/2 1.x*/
153:            for (i=0;i< clrUsed; i++){
154:                    short   r, g, b;
155:                    b = fgetc(fp);
156:                    g = fgetc(fp);
157:                    r = fgetc(fp);
158:                    printf("Color[%3d] = (%3d,%3d,%3d)¥n", i, r, g, b);
159:            }
160:    }
161: }
162:
163: int
164: Get2Bytes(FILE *fp)
165: {
166:    int     ret;
167:
168:    ret = fgetc(fp);
169:    ret += fgetc(fp)*0x100;
170:    return(ret);
171: }
172:
173: int
174: Get4Bytes(FILE *fp)
175: {
176:    int     ret;
177:
178:    ret = fgetc(fp);
179:    ret += fgetc(fp)*0x100;
180:    ret += fgetc(fp)*0x10000;
181:    ret += fgetc(fp)*0x1000000;
182:    return(ret);
183: }
```

各種サウンドデータの解析

CD-ROMはPCMの宝庫

CD-ROM

Taki Yasushi 瀧 康史

CD-ROM内の各種データのうち，サウンドデータに注目しましょう
各種PCM音をX68000で使用できる形式にするにはどうすればいいのか
データ解析を手当たり次第にやってみました

はじめに

割り切って使うCD-ROM。私はPCMにターゲットを絞って使ってみることにしました。音楽データを作るために，音源ソースとしてCDやCD-ROMに入ったPCMデータを吸い出して利用する。なかなかオツというものです。著作権がらみのこともあって，フリーのPCMデータ以外で音楽データを作成し，それを通信回線などに流すのは違法ですが，個人で楽しむ分には問題ないでしょう。

ま，フリーで供給されているPCMもありますしね。

ゲームCDの中には，声優さんのPCMなども入っていたりして，その筋の人にはウケるかもしれません。こんなのも個人で使うにとどめておかねばならないでしょう。まあSX-WINDOW環境を豊かにするのにはよいのではないでしょうか。私は，いまいちしゃべりまくるウィンドウシステムは好きじゃないんだけど。

本文中，「ダンプする」という言葉が出てきますが，これはCDからデータを吸い出すことを意味しています。私ローカルの単語です。CD2PCM.Xやmaddump.xを使ってPCMを吸い出す行為は，吸い出すというよりもデータをダンプすることに似ているので私はこう呼んでいます。

PCMのデータ形式について

まず最初にPCMのフォーマットについて，いろいろ説明していきましょう。

PCMというとひと言ですが，PCMそのもののデータ形式にはいろいろあります。あちこちで，それ相応にフォーマットが違うのでX68000で利用するにはコンバータを利用せねばなりません。

このような複数のPCMをある程度自動的に判別し，勝手にX68000のMSM6258形式のAD PCMに変換してから再生するプログラムを作ってみました。pcmplay.xというプログラムです。長い時間をかけて作成したソフトウェアなので，プログラムが肥大してしまい，リスト掲載は不可能な量になってしまいました。次の付録ディスクまでには，さらに多機能にバージョンアップして，入れるつもりです。乞うご期待！ってとこかな？

今回，このpcmplay.xのデータコンバータの部分だけ，掲載することになったので，興味のある人は参考にしてください。

それでは，各PCMのデータ形式の説明です。

●AD PCM

AD PCMというのは，予測付き差分PCMです。この予測に関する「重み」を，対数グラフによって求めているようです。

この予測に関する関数が違うことによって，同じ4ビットPCMでも少しずつ音が違っているようです。X68000ユーザーにいちばん知られているAD PCMは当然本体内蔵AD PCM音源であるMSM6258のAD PCMでしょう。ほかには，YM2608，YM2610に内蔵されたAD PCMなどがあるようです。

MSM6258のAD PCMは拡張子PCM。pcmplay.xは当然のように対応しています。現時点においては，YM26xx系のAD PCMは非対応です。

また，PCエンジンのCD-ROMに搭載されているPCMもAD PCMで，MSM6258に近いのか(シードが多少違うのかもしれませんが)，データの上下4ビットを入れ換えるだけで綺麗に再生ができます。この拡張子はPCEです。

●16ビットPCM(BigEndian)

Z-MUSICのディスクパッケージに付属している拡張子P16形式のファイルがそれです。BigEndianというのはいわゆるモトローラフォーマットというやつで，アドレスが若い順に上位の桁が格納される方式です。

X680x0のCPUはモトローラCPUですから，当然この形式です。データは符号付きの16ビットで，C言語でいえば，signed short intといったところです。中心は0x0000で，最小値は0x8000(−32768)，最大値は0x7FFF(＋32767)です。

pcmplay.xではP16以外にもPHLという拡張子も許しています。

●16ビットPCM(LittleEndian)

上のLittleEndian版，いわゆるインテルフォーマットというやつです。PC-9821系の内蔵PCM，というよりWindowsのWAVフォーマットの16ビットステレオというのがこれです。基本的にはP16とエンディアンが違うだけなので，符号付きになります。pcmplay.xは内部でエンディアンコンバータを通して，これを再生しています。拡張子はPLHです。

●8ビットPCM(符号付き)

C言語でいえばsigned charで表せるデータです。中心が0x00で，最小値は0x80(−128)，最大値は0x7F(＋127)にあたります。

PC-9821系のPCMで，8ビットモードを使うとこのPCMが使われます。拡張子はPC8です。

●8ビットPCM(符号なし)

C言語でいえばunsigned charで表せるデータです。中心が0x80(＋128)で最小値は0x00，最大値は0xFF(＋255)になります。WAVフォーマットの8ビット版モノラルがこのフォーマットです。ほかには私が以前PPIボードのレビューのときに利用したIC，AD558JのPCMもこれですし，あのPOLYPHONのPCMもこれです。8ビットですから，周波数を上げても低音が出にくいという欠点がありますが。

拡張子はPN8です。

●8ビットPCM(符号ビット付き)

PC8とどう違うかというと，符号のある

なしをMSBによってチェックしている点が違います。C言語で記述される形式はありません。中心点は0x00(+0)，0x80(−0)のいずれかです。最小値は 0xFF(−127)で最大値は0x7F(+127)になります。

FM TOWNSが初期型のときから搭載していたPCMがこれです。ほかにもメガCDのPCMもこれですから，割とデータの数は多くあります。

拡張子はPS8です。

●その他

その他，符号なし16ビットなども考えられるのですが，いまだかつて，符号なし16ビットPCMデータ(エンディアン違い2種類とも)見たことがありません。

したがってpcmplay.xでは対応していませんが，これだけ対応していれば，困ることはないでしょう。

ISO9660のCD-ROM

それでは実際にデータを吸い出してみましょう。もっとも簡単なのはCD-ROMのフォーマットがISO9660形式です。ファイルフォーマットに制限があるのでHuman68kでも見られるのが，このCD-ROMのよいところ。いわゆる標準フォーマットというやつです。ディレクトリが取れるので，PCMっぽいものが，おおよそファイル名で予測できます。これだ！ と思うものを取り出して再生することは容易です。あとは再生ツールさえなんとかすれば，OKというわけです。当然ながら機種によってPCMフォーマットが違いますが，それさえ押さえれば，あとはもうパラダイスです。

●Windows用CD-ROM

Windows用CD-ROMにPCMとして明示的に入っているものは，たいていWAVファイルです。まあWindowsで標準になっているPCMフォーマットですから，それは当然といえば当然ですね。

WAVフォーマットにはヘッダがあり，4つの種類があります。それは，符号付き16bit Littile Endian PCMモノラル(plh)，ステレオ，そして符号なし8ビット PCMモノラル(pn8)，ステレオです。サンプリングレートは44.1kHz，22.05kHzなどが多いようです。利用は簡単な部類に入ります。フリーデータもたくさんありますので音楽屋さんなら，音楽の素材の入ったWindows用のCD-ROMを購入してみるのもいいでしょう。

●AMIGA用CD-ROM

AMIGAのCD-ROMはなかなか日本では手に入らないでしょうが，もともと内蔵音源がPCMのみという背景もあって，音の鳴るソフトなら，まず間違いなく楽器のPCMが入っています。フリーデータには面白いPCM音を使った曲も多く，手に入れれば結構音楽屋さんの素材になるPCMが集まるでしょう。

ファイルとして独立していることはありませんが，音楽データをダンプして見ていればすぐに切り出せるでしょう。内容は符号付き8ビットPCM(pc8)です。サンプリングレートもそれほど高いわけではないので，X680x0の内蔵AD PCMにコンバートするにはよいかもしれません。

●FM TOWNS用CD-ROM

もはや亜AT機となりつつあるTOWNSのCD-ROMですが，古いものと新しいものではタイプが違います。新しいものはWindowsがらみなので，WAVフォーマットである場合が多く，これは上記のとおり簡単に利用ができます。

古いものはSNDファイルといって，TOWNS独自のフォーマットです。ヘッダがあり，フォーマットがやっぱり決まっているので，取り出しも容易です。データのフォーマットとしては，符号ビット付き8ビット PCM(ps8)です。データはモノラルのみ。周波数は約22kHz以下のようです。割と日本では手にしやすいCD-ROMの部類ではないでしょうか。TOWNSという機種は内蔵音源にPCMが多声あるため，PCMデータをシステムディスク(CD-ROM)で供給しています。ときおり，中古でシステムディスクが売られているので購入してみるのも悪いことではないと思います。それほど高いものではありませんし。音楽屋さんには持ってこいのCD-ROMのひとつかもしれません。

●PC-9821用CD-ROM

これももはや亜AT機となりつつあるマシンなので，ときおりWAVファイルで入っていたりします。しかし，PC-9821の内蔵音源やPC9801-86ボードの絡みもあって，WAV以外にもいくつかあります。

まずは，符号付き8ビットPCM(pc8)。楽器としての音はめったに見かけませんが，ないわけではありません。次にYM2608系のAD PCM。MSM6258と対数のシードが違うAD PCMですが，食わせればなんの音かわかる程度には鳴らすことができます。

最近のPC-9821用ソフトはTOWNS兼用である場合が多いので，なかにはSNDファイルで入っていて内部的に変換するものもあるようです。

数が少なく，手に入れても面白いものである可能性は少ないでしょう。

●メガCD用CD-ROM

メガCDはなんとISO9660なんですね。ですからディレクトリを取ることもできます。

PCMは拡張子が特に決まっている様子はありません。ゲームによってヘッダがあったりなかったりしますが，フォーマットはすべて符号ビット付き8ビット PCM(PS8)です。周波数は30kHz未満でしょう。

試しに「ゆみみみっくす」などを見ても，中に入っている＊.BINというファイルはほとんどPS8だったりしますし，「ぽっぷるめいる」なんかは「＊.BIN」がすべてPS8ファイルです(140Mバイトぐらいあったかな？)。もっとも，デモ用の会話ばかりなので，吸い出してもいまいち面白みに欠けますが。

仮に音楽録音されていたにしても，ほとんどがフレーズサンプリングです。どうも，CD-ROMマシンでPCMがたくさんついているマシンは，ほとんどがフレーズサンプリングしている傾向があります。

サンプリングの是非

古くはサンプラーの普及により，そして最近はコンピュータをはじめとしたデジタル楽器類の普及により，音楽関係の著作権問題はそれまでとは少し違った方向にも発展しつつあります。

焦点となるのはサンプリングです。プロのミュージシャンにしても，明らかに他人の出した曲をサンプリングして「素材」として使っている曲を出していることもあります。はなはだしいのになると，フレーズサンプリングで遊んでいるだけで，曲自体はつまらないものなどもありますから，安易なサンプリングでは好感を持って迎えられることは稀なようです。

しかし，なかには音楽的に優れたものも現れ始め，ひとつの音楽ジャンルを形成する勢いを見せるものもあります。こうなると一概に禁止することもできなくなってきています。

また，サンプリング系の楽器のなかには他社の楽器の名前のついた音色があったりもしますから(話がついてるのかな？)，音色の扱いに関しては調べれば調べるほど混乱が広がっていきます。

コンピュータで使う場合にも，最低限加工して使うというのがマナーという考え方や個人で使用する分にはかまわない，音色に著作権はないというものまで，実にたくさんの考え方があります。パソコンの標準機能化しているPCM音源でも，もっとこういう議論もされていいはずなのですが。

まあ，私は結果主義なのでこれで音楽がよければ文句はいいませんけど，CD-DAで直接再生するのに賛成派です。オーケストラ生演奏も入れられますし。

手持ちのメガCDのディスクのうち，パーカッションなどの楽器音がそのまま録音されていたのは，ソニックCDぐらいのものでした。音楽屋にはソニックCDは買いかもしれません。

●SATURN用CD-ROM

なんという早い情報……ってこれをいつ書いているかばれてしまいますね。このCD-ROMもISO9660です。

解析できるCD-ROMがVF(バーチャファイター)しかなかったために，そのほかのジャンルがあるかもしれませんが，VFを見た限りでは8ビット符号なしPCM(*.PN8)といったところです。サンプリングレートがなぜか低く，VFでは6.4kHz程度でした。PCM用のメモリが狭いんでしょうか？ それとも，最初に出るゲームということもあって，ロード時間が長いとイメージが悪くなるとかそういう配慮もあるのか。(編注：第2ロット以降ではサンプリング周波数が変更されているようです)

●NEO・GEO-CD用CD-ROM

NEO・GEO-CDのCD-ROMもISO9660だったりします。フォーマットはAD PCMです。YM2610という，YM2608の変形したようなFM音源チップがついているハズなので，おそらく対数シードはPC-9821のAD PCMとほぼ同一のものでしょう。

拡張子が決まっているようで，PCMは*.PCM。X68000の標準PCMの拡張子とブッキングしてしまいました。実にすっきりとしたCD-ROMで，なにがなにをしているかすぐにわかったりもします(メガCDも割とすっきりしてるけど)。

●プレイディア用CD-ROM

CDDEV.SYSで覗くことはできませんでしたが，CD-ROM.Xでは覗けるようです。一応ISO9660かもしれません。

たまたま編集部にあったプレイディアのCD-ROMを覗いたところ，660Mバイトぐらいのでっかいファイルがひとつ。音声はおそらく映像と一緒に圧縮されていますから，多分，利用価値はないでしょう。

そのほかのCD-ROM

世の中みんなISO9660ならとっても簡単なんですが，そうは問屋が卸さないCD-ROMもいくつかあります。まぁ，解析屋さんの気持ちからいえば，こういうのがあってこそ腕がふるえるのですが……。

●サンプラー用のCD

5ケタぐらいのお値段で，すっごく高いのですが，世の中にはサンプラー用のCDというものが転がっています。これは，サンプラーにとってはCM-64のカードみたいなもので，音色がそのまま録音されて入っているものです。ですからデータとして吸い出すには持ってこいのCDで，値段の分だけ間違いなく働いてくれるはずです。

CD-ROMではなく，音楽CDである場合が多いので，データとして読み取ることはできません。ですから普通のCDプレイヤーにかければ再生できる場合がほとんどです。ということは11月号のフリセレのページで紹介したCD2PCM.xを利用して，そのまま音楽トラックをPCMにコンバートすればおしまいですね。

ノイズも乗らずにきちんとコンバートできますから，音楽野郎にはぴったりでしょう。私もこれでいくつかPCMをダンプして，自作音楽データにいつか使ってやろうかなあなんて思っています。ところがデータを作る暇がないんだな。最近いろんなことやりすぎて……。

●MachintoshのPCM

かなりいろんなところに落ちているCD-ROMなので有効に利用できれば越したことはありません。残念ながら身の回りに新しいCD-ROMしかなかったので，古いものを探すことができませんでした。新しいものは，たいてい44.1kHzの符号付き16ビットPCMBigEndianモノラル(p16)のようですから，取り出すことができれば再生は容易です。zvtで読み込むこともできますね。

問題は取り出しで，ファイルそのものは，MacFileTransfer.xや，CDROM.Xを利用すればOKなのですが，肝心のデータの中身は拡張子RESのリソースファイルにアーカイブされていたり(要はまとめてあるということ)するので，簡単にはいきません。

どこにサウンドファイルがあるかは，ちょっと根性で探すしかないかもしれません。膨大なCD-ROMのなかから探すのは，結構きつい作業です。そもそもサウンドが入っていそうなCD-ROMを探すことも割と難しいので，結構冒険の類になってしまうかもしれません。底辺が広いので，おそらくサウンド用CD-ROMとして売っているものもあるんでしょうけど。

●PCエンジンのCD-ROM

おそらく日本でいちばん手にしやすいCD-ROMがこれでしょう。秋葉原にはドサッとありますし，地方のゲームショップなどにいってもたいてい手に入ります。

とはいえ中身は利用価値があるかどうか謎なものばかり。ナレーションやオープニングデモが音楽CDとしてダイレクトに録音されていたり，データの中に埋まっていたりさまざまです。まあ，こういうCD-ROMを持っている人は，そういう類の人なのでしょうから，吸い出せば楽しいものなのかもしれません。

音楽CDとして録音されている場合は，CD2PCM.Xを利用してダンプします。1トラック目には「このデータはPCエンジンのCD-ROMディスクです。2曲目に……」というメッセージが録音されています。そのゲームがキャラゲーだったりすると，そこに登場する声優さんたちを使ってなんらかの会話を吹き込んでいる場合もあったりしますから，吸い出して楽しい人はCD2PCMでダンプしましょう。

問題はデータトラックの中にPCMが埋め込まれている場合です。PCエンジンのCD-ROMは特殊なフォーマットのため，そのままでは読み出すことはできません。独自にダンプしてみたところ，データにFATらしきものはなかったので，おそらくベタでそのまま記録されているのでしょう。

そうなってくると方法はひとつ。2曲目のコンピュータデータを直接専用ソフトでダンプして，なかからPCMと思われる部分を抜き出すしかありません。ダンプには，拙作のPCEngineCDROMDump.xを利用してみました。CD2PCMはオーディオトラックを読み込むツールなので，ここでは使えないのです。

PCエンジンのPCMデータフォーマットはAD PCMです。かなりX680x0のPCMフォーマットと似ていて，周波数も16kHz,8kHzなどがあるようです。直接X680x0のMSM6258に食わせても，判別できる程度に音はなりますが，これでは多少ノイジーなのでちょっとした小細工をします。

CD-ROMに記憶されているデータは相変わらず膨大です。この中からPCMだと思われる部分をサーチせねばならなく，規模の大きなゲームでは気の遠くなる作業です。なにしろ，聞かなくてはわからないのですから。

そこで，ちょっとしたAD PCMの特質を生かして，特殊なツールを作ってみました。どうするかを教えてあげたいところですが，その筋の人が対策してしまうと悲しいから，ここではないしょ。

効果はてきめんで，PCエンジンのディス

クをCD-ROMドライブに挿入して，拙作のmaddump.xを使うだけで，CD-ROMに記録されているデータのうちPCMではないかと思われる部分をザクザク個別ファイルに落としていきます。多少時間がかかるので，CD-ROMドライブにPCエンジンのCD-ROMを入れたら，さっさと出かけるなり，寝てしまうなりします。

どうしてmaddumpなのか。

それはもうPCエンジンのCD-ROMって声優さんの声ばっかりだから。プログラム作成のために，テストダンプして聞いていたら頭がウニになってきそうでした。このツールを使う人って多分maddness。

まとめ

リスト1はC言語用のPCMコンバータです。性格上，C言語をまったく使用できない人には全然役に立ちません。そういう方は次の付録ディスクを待ってください。

また，リスト中はできる限りコメントを入れたつもりです。気分でコメントの数が変わる人間なので，多かったり少なかったりしますが，呼び出し方法は関数の最初に記述されているため，おそらくわかると思います。

音声データや音楽用素材データの吸い出し用と割り切って遊ぶだけで，CD-ROMドライブの価値はあると思っています。音楽屋さんにはCD-ROMは必携ではないかと思うのですがいかがでしょう。もちろん，CD2PCMの存在なくしてはこれを語ることはできません。「できる限りこれの動作するソニー，東芝ドライブを！」といったところです。

さて，余談ですがCD2PCMとシングルCDによくカップリングされているカラオケ演奏を利用して，ボーカルだけを抜くプログラムができています。もちろん，完全に個人で遊ぶためのアイテムです。ちょっとまずいかな？　って感じもしますけど，これも機会があったら公開しましょう（まだバグがあるので）。

しかし，これだけPCMが揃ってくると，データ保存場所もかなり必要になってきます。MOは必須，できれば230MバイトMOもほしいかもしれません。そして，思うのが高音質PCMボードの存在。音質はAWESOME-Xを使うのがNo.1でしょうけど。自作している PCMボードは，少し難航するところがありまして（サブCPUやグラフィック転送機能関連で）まだ先になりそうです。形になったら公開しますが，はたしていつになることやら。

サンプリングレートについて

最近はPCM音源といえばCDクオリティ＝16ビット44kHzを示すようになっています。

CDのサンプリングレートが人間の可聴範囲から決められたというのは有名ですが，CDの登場当時から非CD派の人たちは，高音域の音も「音」としては認識できなくても別の刺激として知覚できる，という主張をしていました。しかし，たった4kHz分サンプリングレートの高いDATの48kHzサンプリングについては悪口をいう人は見たことがありません。

X68000の音声を聞けばわかるように，人間がしゃべっている声あたりではそれほど高い周波数は必要とはしません。ゲームなどでは8kHzあたりを使っているものものもたくさんありますが，それほど不自然に聞こえることはないでしょう。これはX68000のAD PCM音源が人間の声をサンプリングすることを対象に設計されていることにもよります。楽器音などには弱いはずなのですが，時として出るはずのない音成分まで再現しているような錯覚を起こすこともあります。そう思えば，実際にはそれほど高いサンプリングレートは必要ないのかもしれません。ちょっと前まで楽器音には12ビットで20k～30kHzのものがプロ用として立派に通用していましたし……。

ゲームのなかには8kHzなのに非常にクリアなサンプリング音が使われているものもあります。逆に16ビットサンプリングなのに音が悪いということもあります。結局はデータの作り方次第ですから，周波数が高いものが即，高音質とは限りません。高忠実性を持っていることは確かですが。

リスト1

```
 1: /*      converters.c
 2:
 3:        このプログラムは、pcmplay.x内部で行う、
 4:        ＰＣＭデータフォーマットコンバータ部分が記述されています。
 5:        ソーストディスティネーションが分かれている
 6:        プロシージャの場合、ディスティネーションはそれに記す分の
 7:        データを予め確保しておくこと。
 8:
 9:  $Id: converters.c,v 1.2 1994/08/31 23:12:30 Kohju Exp $
10:  $Header: a:/usr/home/Kohju/Labo/PCMPLAY/converters.c,v 1.2 1994/
08/31 23:12:30 Kohju Exp $
11:  $Log: converters.c,v $
12:  * Revision 1.2  1994/08/31  23:12:30  Kohju
13:  * ・リヴィジョンチェックルーチンの追加。
14:  *
15:  * Revision 1.1  1994/08/31  19:51:58  Kohju
16:  * Initial revision
17:  *
18:
19: */
20:
21: #include        <stdio.h>
22: /* #include     "pcmplay.h" */
23:
24:
25: void    pn8topc8(void *p,int len)
26: /*      pn8からpc8への変更
27:         *pにアドレス、lenに長さを入れて下さい。
28:         ソースデータがそのまま書き変わるので、
29:         元のデータを保存する必要がある場合は、
30:         自分で保存しておくこと。
31: */
32: {
33:         unsigned char   *cp=p;
34:                         int     i;
35:
36:         for(i=0;i<=len-1;i++,cp++)
37:                 (char)*cp = (char)(*cp-0x80);
38: }
39:
40: void    ps8topc8(void *p,int len)
41: /*      ps8からpc8への変更
42:         *pにアドレス、lenに長さを入れて下さい。
43:         ソースデータがそのまま書き変わるので、
44:         元のデータを保存する必要がある場合は、
45:         自分で保存しておくこと。
46: */
47: {
48:         unsigned char   *cp=p;
49:                         int     i;
50:
51:         for(i=0;i<=len-1;i++){
52:                 if( *cp >= 0x80 ) *cp = 0x80 - *cp;
53:                 cp++;
54:         }
55: }
56:
57: void    ps8topn8(void *p,int len)
58: {
59: /*      ps8からpn8への変更
60:         *pにアドレス、lenに長さを入れて下さい。
61:         ソースデータがそのまま書き変わるので、
62:         元のデータを保存する必要がある場合は、
63:         自分で保存しておくこと。
64:
65:         AD558のデータフォーマットは$00-$ffの線形である。
66:         ps8形式は、MSBがサインビットの8bitPCMである。
67:
68:         よって、データが＄７Ｆ以下の時、
69:                 データは＄８０もち上げられる。
70:         逆に、  データが＄８０以上の時、
71:                 データは＄８０から引かれ、（－符号が付く）
72:                         ＄８０持ち上げられる。
  */
73:
74:         unsigned char   *cp=p;
75:                         int     i;
76:
77:         for(i=0;i<=len-1;i++){
78:                 if( *cp >= 0x80 ) *cp = 0x80 - *cp;     // 符号を
つける。
79:                 *cp += 0x80;    // 80持ち上げられる
80:                 cp++;
81:         }
82:         return;
83: }
84:
85: void    lh_bitconv(void *p2,void *p,int len)
86: {
87: /*      PCE形式のPCMファイルを、X68kのPCM形式にデコードする
88:         *pにソースアドレス、*p2にディスティネーションアドレス、
89:         lenに長さを入れておくこと。
90:         ディスティネーションとして*p2から始まるメモリは
91:         ソースと同じくlenと同じでよい。
92: */
93:
94:         char    *cp =p;
95:         char    *cp2=p2;
96:         char    temph=0;
97:         char    templ=0;
98:         int     i;
99:
```

▶ワーイ。ついにX68030のPOPスタンドを手に入れました。　関　喜視(21)神奈川県

```
100:         for(i=0;i<=len;i++,cp++,cp2++){
101:                 templ=((*cp & 0x0f) << 4);
102:                 temph=((*cp & 0xf0) >> 4);
103:                 *cp2=(temph | templ);
104:         }
105: }
106:
107: void    pc8topn8(void *p,int len)
108: {
109: /*      ps8からpc8への変更
110:         *pにアドレス、lenに長さを入れて下さい。
111:         ソースデータがそのまま書き変わるので、
112:         元のデータを保存する必要がある場合は、
113:         自分で保存しておくこと。
114:         (unsigned char)から-0x80して、(char)にキャストする。
115: */
116:         unsigned char   *cp=p;
117:                         int     i;
118:
119:         for(i=0;i<=len-1;i++){
120:                 *cp = (char)((*cp)- 0x80);
121:                 cp++;
122:         }
123:         return;
124: }
125:
126: void    endianconv(void *p2,void *p,int len)
127: {
128: /*      エンディアンをコンバートする。
129:         *pにソースアドレス、*p2にディスティネーションアドレス、
130:         lenに長さを入れておくこと。
131:         ディスティネーションとして*p2から始まるメモリは
132:         ソースと同じくlenと同じでよい。
133: */
134:
135:         short int       *cp =p;
136:         short int       *cp2=p2;
137:         short int       temph=0;
138:         short int       templ=0;
139:         int     i;
140:
141:         for(i=0;i<=len;i++,cp++,cp2++){
142:                 templ=((*cp & 0x00ff) << 8);
143:                 temph=((*cp & 0xff00) >> 8);
144:                 *cp2=(temph | templ);
145:         }
146: }
147:
148: void    pc8top16(void *p2,void *p,int len)
149: {
150: /*
151:         符号付き8bitPCMを符号付き16bitPCMに変換する。
152:         p2に出力されるpcmのアドレス（16ビット）関数を呼ぶ時に、
153:         lenの2倍の長さを確保しておくこと。
154:       · pに入力されるpcmのアドレス（16ビット）
155:         lenは入力されるpcmの長さ。
156: */
157:         char            *cp =p;
158:         short int       *cp2=p2;
159:         int     i;
160:
161:         for(i=0;i<=len;i++){
162:                 *cp2=(short int)((*cp)*8);
163:                 cp++;
164:                 cp2++;
165:         }
166: }
167:
168:
169: void    p16topc8(void *p2,void *p,int len)
170: {
171: /*      符号付き8bitPCMを符号付き16bitPCMに変換する。
172:
173:         p2に出力されるpcmのアドレス（16ビット）関数を呼ぶ時に、
174:         lenの2分の1の長さを確保しておくこと。
175:         pに入力されるpcmのアドレス（16ビット）
176:         lenは入力されるpcmの長さ。
177:         この関数はマキシマムレンジを操作していないため、
178:         いまいち未完全ではあります。
179: */
180:         short int       *cp =p;
181:         char            *cp2=p2;
182:         int     i,s=8;
183:
184:         for(i=0;i<=len;i++){    // s倍という敷居値だな。それ以上
は音が割れると。
185:                 if(*cp> 1023) *cp = 128*s-1;
186:                 if(*cp<-1024) *cp =-128*s;
187:                 *cp2=(char)((*cp)/s);
188:                 cp++;
189:                 cp2++;
190:         }
191: }
```

リスト2

```
  1: * 周波数を変換するプログラム(c) Z.Nishikawa,modified by Kohju
  2: *
  3: *   引き数)
  4: *   ソースレングス(0)
  5: *   入力周波数(1)
  6: *   出力周波数(2)
  7: *   ディスティネーション(3)
  8: *   ソースアドレス(4)
  9:
 10: *   返り値)
 11: *   ディスティネーションの長さ。(6)
 12:
 13: * int   freqconv( int dsize, int sfreq,int dfreq,void *daddr,voi
d *saddr);
 14:
 15:         .xdef   _freqconv
 16:
 17:         .offset 4
 18:
 19: dsize   ds.l    1       * (0)
 20: sfreq   ds.l    1       * (1)
 21: dfreq   ds.l    1       * (2)
 22: daddr   ds.l    1       * (3)
 23: saddr   ds.l    1       * (4)
 24:
 25:         .text
 26:
 27: savesize        equ     4*9
 28:
 29: _freqconv:
 30:         movem.l d3-d7/a3-a6,-(sp)
 31:
 32: *       全てのアドレスは使用可能
 33:
 34: D5にはソースサイズ:
 35:         nop
 36: D6にはソース周波数:
 37:         nop
 38: D7にはディスティネーション周波数:
 39:         nop
 40: A1にはディスティネーションアドレス:
 41:         nop
 42: A2にはソースアドレス:
 43:         movem.l dsize+savesize(a7),d5-d7/a1-a2
 44:
 45: do_ajfr:
 46:         * < d5.l=data count                     (0)(Datasize/2)
 47:         * < d6.w=source frq                     (1)
 48:         * < d7.w=destination frq                (2)
 49:         * < a1.l=destination pcm data address   (4)
 50:         * < a2.l=source pcm data address        (3)
 51:         * > d0.l=destination pcm data size      (6)
 52:         * X d0-d7
 53:
 54:         lsr.l   #1,d5           * /2
 55:         pea     (a1)
 56:         exg.l   d6,d7
 57:         divu    d6,d7
 58:         move.w  d7,d1           *d1=step
 59:         clr.w   d7
 60:         divu    d6,d7
 61:         swap    d7
 62:         tst.w   d7
 63:         beq     @f
 64:         add.l   #$0001_0000,d7
 65:         clr.w   d7
 66: @@:
 67:         swap    d7                      *d7=revise
 68:         moveq.l #0,d3
 69:         tst.w   d1
 70:         bne     nz_d1
 71: doa_lp00:
 72:         move.w  (a2)+,d0
 73:         add.w   d7,d3
 74:         bcc     @f
 75:         move.w  d0,(a1)+
 76: @@:
 77:         subq.l  #1,d5
 78:         bne     doa_lp00
 79:         bra     exit_doa
 80: nz_d1:
 81:         move.w  d1,d6
 82:         move.w  (a2)+,d0
 83:         move.w  (a2),d2
 84:         cmpi.l  #1,d5                   *最後かどうか
 85:         bne     @f
 86:         move.w  d0,d2
 87: @@:
 88:         add.w   d7,d3
 89:         bcc     @f
 90:         addq.w  #1,d6
 91: @@:
 92:         ext.l   d0
 93:         ext.l   d2
 94:         movem.l d1/d3,-(sp)
 95:         sub.l   d0,d2
 96:         divs    d6,d2
 97:         move.w  d2,d1                   *d1=step
 98:         clr.w   d2
 99:         divu    d6,d2
100:         swap    d2
101:         tst.w   d2
102:         beq     @f
103:         add.l   #$0001_0000,d2
104:         clr.w   d2
105: @@:
106:         swap    d2                      *d2=revise
107:         moveq.l #0,d3
108: doa_lp01:
109:         move.w  d0,(a1)+
110: @@:
111:         add.w   d1,d0
112:         add.w   d2,d3
113:         bcc     @f
114:         tst.w   d1
115:         bpl     doa_pls
116:         subq.w  #1,d0
117:         bra     @f
118: doa_pls:
119:         addq.w  #1,d0
120: @@:
121:         subq.w  #1,d6
122:         bne     doa_lp01
123:         movem.l (sp)+,d1/d3
124:         subq.l  #1,d5
125:         bne     nz_d1
126: exit_doa:
127:         sub.l   (sp)+,a1
128:         move.l  a1,d0
129:         movem.l (sp)+,d3-d7/a3-a6
130:         rts
```

▶「デイトナUSA」のおかげで，自転車に乗っているとついライン取りを考えてしまう。
伴　武士(23)千葉県

SCSI2でのCDデバイス使用法

オーディオCDを再生する

CD-ROM

Taki Yasushi 瀧 康史

CD-ROMはちょっと特殊なSCSI機器です
単なるデータ媒体としてだけでなくさまざまな側面を持ちます
ここではSCSIコマンドを使って音声を再生してみましょう

X680x0でCD-ROMを使うとすれば，やっぱりソフトを作んなきゃいけませんよね。実際,CD-ROMをソフトで扱うにはそれ相応に苦労するわけで，そのための資料も結構必要です。手順を覚えるのもひと苦労。

そこで……「完全無欠！ X68000で操作するSCSI2 CD-ROM解説」というわけにはいかないのですが，とりあえず，その筋の本を見る前に必要な予備知識をここで解説することにしましょう。

概要

SCSI1ではCD-ROMは「リードオンリーデバイス」として扱われています。ゆえにCD-ROM独特の音声/映像データなどの扱いが詳しく決められていませんでした。

しかしオーディオデータが扱えないというのは商品的に見ても大きな欠点です。その結果，SCSI1時代のCD-ROMデバイスはベンダ固有コール(単一メーカーだけのコール)を利用してオーディオデータを取り扱っていました。問題になるのは，想像どおりメーカーごとの互換性です。はっきりいってこの時代のCD-ROMドライブは絶対に買ってはいけません。メーカーによって取り扱いが全然違うからです。

CCS案（要するにSCSI2とSCSI1の間です)を経てSCSI2になるとき，こういった背景からか，SCSI2ではリードオンリーデバイスがCD-ROMデバイスと呼び変えられました。ここで初めてCD-ROMがCD-ROMとして規格上認められ,「オプション」ですがオーディオを操作するファンクションの取り決めが行われたのです。

これによって,CD-ROMドライブの操作がある程度標準化され，SCSI2対応のCD-ROMドライブならプログラムから同じ手順で操作できるようになりました。標準化が遅すぎたため,X680x0ではサポートされていない機能です。CD-ROM操作に必要そうなCSIコールもありませんし，唯一標準といえる計測技研のデバイスドライバ，CDDEV.SYSは多機種での実験をしていないためか，ときおり不具合が見え隠れします。

いちばん「嫌」なのが，オーディオ再生機能です。SX-WINDOW上でCD-DA(CDの音声部分）を再生するCDPlayer.Xというソフトが計測技研のCD-ROM DRIVER ver.2.00に付属しています。しかもライブラリまで入っていて，C言語さえ使えればあなたにも簡単にCD-DAが再生できますよ！ といわんばかりなのですが，これが東芝ドライブ以外ではうまく再生できないんですよ。ほかにも動くドライブがあるのかもしれませんが，少なくともソニーCDU-55Sユーザーの私は再生ができませんでした（注：最新版のドライバでは可能)。

自作ソフトでCD-ROMドライブを扱うときに必要なのは主にCD-DAの再生です。同人ソフトなんかでゲームのBGMにCDを利用するのもいいかもしれません。そこで計測技研のドライバにはこういうライブラリが入っているのでしょうが，ソニードライブで動かないのは少し問題ですね。

一般にCD-ROMデバイスは，IEC標準の，いわゆるCDを読み出すことができます。このCDは大きく分けて，デジタルオーディオ用のCD-DA(Digital-Audio)と，デジタルデータ専用のCD-ROM,そしてこれらが交ぜられたものがあります。交ぜられたものもCD-ROMといいます。

CD-DA部とデータ部ではきちんした違いがあり，これらは完全に分けられています。オーディオ機器のCDプレイヤーでCD-ROMを再生した場合，CD-ROMのデータ部を再生できますが,CD-ROMドライブでは普通，データ部は再生できません。また,SCSI規格によって用意されたREADコマンドでCD-ROMのデータ部は読めますが,オーディオデータは本来読むことはできません（CD2PCM.Xでサポートされている CD-DA部のリード機能は一部メーカーの独自のもので，ほかのメーカーのドライブを買った場合にはどうしようもない)。

一般にCD-ROMのCD-DA部を再生する手順はSCSI2によって決められているので,SCSI2に対応したドライブなら，同じ手順で再生できるはずです。そこで,SCSI2準拠のCD-ROMドライブでCD-DAを再生するライブラリを作ることにしました。

演奏までの手順

すでに話したとおり，X680x0のSCSIコールにはCD-ROM関連のファンクションは用意してありませんからコマンドアウト(XC _S_CMDOUT,libc _scsi_cmdout)で直接SCSIコマンドを書き出して実行しなくてはならなくなります。

演奏手順を説明しましょう。最初にTEST UNIT READYコマンドを実行します。これはすべてのデバイス共通の命令でSCSI2でのサポートは必須になっています。X680x0では_S_TESTUNIT(XC)かscsi_testunit(libc)で行います。この命令で，SCSI機器がきちんとつながっていて，かつイニシエータからのコマンド待ちの状態かをチェックします（つまりBUSYチェック)。

ターゲットが暇であるなら，次にINQUIRYを発行します。この命令は接続されたデバイスはなにかと質問する命令です。SCSIデバイスにはいろいろありますから，これによってそのターゲットが，CD-ROMであるかをチェックします。

そして，次にドライブに入ったCDの情報を得なくてはなりません。これにはReadTOCという命令を実行します。

このコマンドは，CD-ROMのTOC(Table Of Contents)領域を読み込み，イニシエータに送る命令です。このTOC領域

には，そのCD-DAの最始トラック，最終トラック，最終演奏時間から，その論理アドレス，指定されたトラックの通算開始時間，属性(データトラックなのかオーディオトラックなのかまで)などが記載されています。

実際のCDの演奏を担当する命令はなぜか6種類もあり，命令の長さが違っていたり，引数が違っていたりさまざまです。要所要所で使いやすい奴を選べばよいというわけですか。おそらく，SCSI2のCD-ROMドライブでオーディオをサポートしているドライブなら，これらはすべてサポートされているでしょうから，オプション命令でも心配せずに使用してよいでしょう。

CD再生のコマンドを以下に示します。

●PlayAudio(10)

演奏開始場所を論理ブロックアドレスで指定し，演奏する長さをデータ長（論理ブロック数）で指定します。論理ブロックは2048バイト単位で管理されています。データ長が2バイトしかないため，長さは最大65535しかありません。というわけで実用的ではない命令なのですが，命令長が短いという利点があります。

●PlayAudio(12)

上の命令が12バイト命令になり，再生データ長が4バイト長になったものです。

●PlayAudioMSF

演奏開始時間，演奏終了時間をアブソリュートMSFアドレスで指定できるものです。アブソリュートというのはCD-ROM先頭からの絶対番地を表しています。MSFというのは，Mが分，Sが秒，Fがフレーム(1/75秒単位)を表しています。要するに演奏時間によるアドレス指定です。

●PlayAudioTrackIndex

演奏開始トラック，演奏終了トラックで指定する再生命令です。

●PlayAudioTrackRelative(10)

この命令は，トラック相対の論理ブロックアドレスで演奏開始場所を指定し，データ長で演奏する長さを指定します。例によって演奏する長さは2バイトなので実用的な命令ではありません。

●PlayAudioTrackRelative(12)

上の命令の再生データ長が4バイト長のものです。

*　　　*　　　*

大きく分けて3種類。演奏開始場所を論理アドレスで指定するか，MSFアドレスで指定するか，トラックで指定するかです。

ReadTOCコマンドの戻り値は，引数によりMSFか論理アドレスで返ってきます。演奏最始トラック，最終トラックは常に戻り値としてあります。

楽なのはトラックで指定する方法です。サンプルプログラムではPlayAudioIndexで再生していますが，トラックで指定すると細かい指定ができませんので，サンプルでは使っていませんが，アブソリュートMSFアドレスで指定できる，PlayAudioMSFも関数として用意しておきました。

CDB形式

実際にCD演奏するための手順はわかりましたね。そのためのSCSIコマンドを規格書で見ると，CDB形式で記載されています。このCDB形式のコマンドを転送するための手段を解説しましょう。

1)　アービトレーション&データフェーズを実行

このコマンドにより，イニシエータがターゲットにデータを転送するためのバスラインを獲得します。やらないとイニシエータとターゲットとの間に道ができません。要は，どのIDを持ったターゲットがこれから送るイニシエータの命令を受け取るか，砕いていえばIDの指定ですね。

命令は，_S_SELECT(XC形式)か，_scsi select(libc形式)で行います。

ちなみに，アービトレーションフェーズは，マルチイニシエータシステムのときに，アクセスがかちあってしまうのを防ぐためにあります。ちなみに，X680x0のSCSIドライバはこれを行っていないため，2つのイニシエータでターゲットを共有することはできません。

もっとも，これだけしても同時にひとつのターゲットを利用するためには，OS内部のキャッシュなど，ソフト的にクリティカルな部分が多数出てくるので，すぐにマルチイニシエータシステムが実現するわけではありません。念のため。

2)　実際にCDB形式でコマンドを出力

コマンドの出力は_S_CMDOUT(XC表記)か，_scsi_cmdout(libc表記)を実行します。実行命令は長さと，開始アドレスがこの命令の引数なので，データは配列変数に入れることになります。

PlayAudioMSFのように，送っておしまいな命令はこれで終わりです。

3)　戻り値のデータを受け取る

戻り値が必要なファンクションの場合，受け取るために，ステータスインフェーズ，データインフェーズを実行します。

ステータスインフェーズはターゲットからイニシエータに1バイト長(255まで)のデータをもらうものです。これ以上のデータはデータインフェーズで受け取ります。

これらの手順をC言語のライブラリにしたのが，_scsi_cmdoutexです。手順はWATA氏のCD2PCMを参考にさせていただきました。命令のフォーマットなどは，ソースを参照してください。

実際に演奏する

CDB形式のコマンドの送り方さえわかってしまえばあとは簡単です。手順に従っ

表1　Play Audio(12)

バイト＼ビット	7	6	5	4	3	2	1	0
0	A5h(オペレーション・コード)							
1	LUN			Reserved				RelAdr
2	(MSB) 論理ブロック・アドレス							
3								
4								
5	(LSB)							
6	(MSB) 転送データ長							
7								
8								
9	(LSB)							
10	Reserved							
11	コントロールバイト							

表2　Play Audio MSF

バイト＼ビット	7	6	5	4	3	2	1	0
0	47h(オペレーション・コード)							
1	LUN			Reserved				
2	Reserved							
3	開始・Mフィールド							
4	開始・Sフィールド							
5	開始・Fフィールド							
6	終了・Mフィールド							
7	終了・Sフィールド							
8	終了・Fフィールド							
9	コントロールバイト							

表3　Play Audio Track Index

バイト＼ビット	7	6	5	4	3	2	1	0
0	48h(オペレーション・コード)							
1	LUN			Reserved				
2	Reserved							
3	Reserved							
4	開始トラック							
5	開始インデックス							
6	Reserved							
7	終了トラック							
8	終了インデックス							
9	コントロールバイト							

表4　Play Audio Track Relative(10)

バイト＼ビット	7	6	5	4	3	2	1	0
0	49h(オペレーション・コード)							
1	LUN			Reserved				
2	(MSB) TRLBA(トラック相対論理ブロック・アドレス)							
3								
4								
5	(LSB)							
6	開始トラック							
7	(MSB) 転送データ長							
8	(LSB)							
9	コントロールバイト							

てコマンドを送れば再生できます。

リスト1はSCSI2 CD-ROMドライブ対応簡易CDDA再生ツールです。とりあえず指定トラックから指定トラックまでの再生，ポーズ，レジュームなどができます。

こちらで確認した限りでは，東芝ドライブ，ソニードライブでうまく動作しました。コパル，メルコCDS-EともにOKです。ロジテックSCD-200のNECドライブは，SCSI2対応のハズなのですが，どうもReadTOCコマンドがサポートされていないのか，うまく再生できません。いろいろもがいてみたのですが，どうもだめですね。

ReadTOCコマンドは基本的にはオプションコマンドなのですが，オーディオコマンドをサポートしたドライブは普通サポートしています。きっとSCSI2から見るとNECドライブはオーディオが扱えないドライブなのでしょう。イヤホン端子があるのになんで……とお思いでしょうが，とりあえずNECだからといっておきましょう。

リストとともに，制作したライブラリを簡単に説明しましょう。

_scsi_ReadTOC_MSFは戻り値をMSF形式に固定してReadTOCを発行する命令です。引数はIDとトラック番号だけで，あとは戻り値を保存するためのポインタです。トラックは基本的に1からなのですが，0を保存すると，CDのM,S,FにCDの最始アドレス（MSF表記）が返ってきます。

また，トラックに0xaaを代入するとM,S,FにCDの最終時間をMSF単位で返します。

minはトラックに0を指定すると最初のトラックが，それ以外では指定トラックと同じものが返ります。CDの最初のトラックは1に決まってると思うかもしれませんが，そうとは限らないそうです。また，maxにはどんな値を指定しても，最後のトラック番号が返ってくるようです。

トラックに適当な値を指定した場合，そのトラックの開始アドレス（MSF単位）が返ってきます。この適当なトラック番号ですが，必ずminからmaxまでの間にしてください。でないとReadTOCを発行する段階で停止します。

本来，ReadTOC命令は，MSF形式以外に論理アドレス形式で返すこともできますが，この関数では省略しています。また，オーディオトラックとデータトラックかの判断もできますが，ここではこれも省略しています（詳しくは参考文献）。

_scsi_PlayAudioMSFは，MSF単位でスタートポイント，エンドポイントを与えて演奏を開始します。基本的にMSF単位はアブソリュート（CD先頭からの時間）です。この関数はプログラム中で利用されていませんが，CD-ROMでゲームなどを作る際，トラックの途中から再生する命令は必要かなということで用意してみました。

実際にサンプルで利用している命令はscsi_PlayAudioIndexです。引数はIDと開始トラック，インデックス，終了トラック，インデックスの5つ。インデックスはトラックをさらに細かくわけたようなものなのですが，こういう形式で録音されているCDはもはや見ませんし，対応しているオーディオCDプレイヤーも見たことがありません。というわけで，1に固定しています。Indexに対応したCDを持っている方は勝手にソースを書き直してください。

最後に_scsi_PauseResumeです。引数はIDとPRでPRにはポーズなら0，レジュームなら1を指定します。CD演奏を途中で停止したいときにポーズをしてください。レジュームはポーズの解除を意味します。

＊　　＊　　＊

たとえばメガCDのゲーム，ゆみみみっくすを実行してしまうプログラムは実際にあります。プログラムはすべてオリジナルでデータだけを利用したというものです。初めてX68000上でゆみみみっくすを見たとき，結構感動しました。まあ，こういったプログラムは個人で作った挙句，人に見せる程度で終わるものですが，こういったプログラムを見せてもらうと，結構ドキドキしますよね（私は持っていないし，編集部にもないので念のため）。

まあ，同人ゲームやそういったものを作るうえで，とあるCD-ROMが必要ですとか，そういうのは悪くないかもしれません。怪しげな地下プログラムを奨励しているつもりではないし，最近のプログラムは「解析を禁ず」と，法的に可能なのだかどうだかわかりませんが，そう書いてあるものもあります。まあ，解析魔にはやりづらい世の中になったかもしれません。

ま，悪いことしないと人間は成長しませんよ。ほっほっほ，と笑っておきましょう。

参考文献

CD2PCM.Xのソース，WATA

OPEN DESIGN No.1 SCSI完璧リファレンス，CQ出版

SCSI-2詳細解説，菅谷誠一，CQ出版

リスト

```
  1: #include <stdio.h>
  2: #include <stdlib.h>
  3: #include <unistd.h>
  4: #include <string.h>
  5: #include <sys/dos.h>
  6: #include <sys/iocs.h>
  7: #include <sys/scsi.h>
  8:
  9: /*
 10:   コンパイルに関して
 11:   gcc cdp.c -O -Wall -fomit-frame-pointer -fstrength-reduce -fforce-mem -fforce-addr -fcomb
ine-regs -finline-functions -lscsi -ldos -liocs
 12:
 13:   環境変数は下の通りです。
 14:   set GCC_OPTION= LFIAMWOE+
 15:   set MARIKO=AB
 16: */
 17:
 18: /*    型定義    */
 19:
 20: struct  __scsi__inquiry {
 21:   unsigned char  Qualifier_DeviceID;
 22:   unsigned char  RMB_DTModifier;
 23:   unsigned char  ISO_ECMA_ANSI;
 24:   unsigned char  AENC_TrmIOP_ResponceData;
 25:   unsigned char  DataSize_4;
 26:   unsigned char  Reserved1;
 27:   unsigned char  Reserved2;
 28:   unsigned char  Support;
 29:   unsigned char  Bender_ID[8];
 30:   unsigned char  Product_ID[16];
 31:   unsigned char  Product_Ver[4];
 32:   unsigned char  Reserved3[220];
 33: }s_inq;
 34:
 35:   void init_inq( void );            // SCSIデバイスの初期化
 36:
 37:   int  _scsi_PauseResume( int,unsigned char );
 38:                                     // Pause & Resume
 39:   void _scsi_ReadTOC_MSF( int,unsigned char,unsigned char *,unsigned char *,unsigned char *
,int *,int *);
 40:                                     // Read Table of Contents
 41:   int  scsi_cmdout_ex(int, int, const void *, int, void *);
 42:   int  _scsi_PlayAudioTrackIndex( int, unsigned char, unsigned char, unsigned char, unsigne
d char);
 43:                                     // PlayAudio
 44: /*    グローバル変数    */
 45:
 46: int  CDROM_ID=6;                    // CDROMのID、デフォルトは6。
 47:
 48: void  main(int argc,char **argv)
 49: {
 50:   char  *env_var;                   // 環境変数会得のためのポインタ
 51:   int    min,max;
 52:   unsigned char SM,SS,SF;
 53:   unsigned char EM,ES,EF;
 54:   int    st=0,et=0,c;
 55:   printf("cdp.x SCSI2準拠CDROM専用簡易CDDA再生プレイヤー by 瀧¥n");
 56:
 57: /*          環境変数          */
 58:   if((env_var=getenv("CDROM"))!=NULL)  CDROM_ID=atoi(env_var);
 59:
 60:   while((c = getopt(argc,argv,"pPrRi:s:e:I:S:E:")) != EOF) {
 61:     switch (c) {
 62:     case 'P':
 63:     case 'p':
 64:       printf("Pause.¥n");
 65:       _scsi_PauseResume( CDROM_ID ,0);
 66:       exit(0);
 67:       break;
 68:     case 'R':
 69:     case 'r':
 70:       printf("Resume.¥n");
 71:       _scsi_PauseResume( CDROM_ID ,1);
 72:       exit(0);
 73:       break;
 74:     case 'I':
 75:     case 'i':
 76:       CDROM_ID = atoi(optarg);
 77:       if ( CDROM_ID < 0 || CDROM_ID > 7 ) {
 78:         printf("SCSI ID が不適当です。¥n");
 79:         exit(-1);
 80:       }
 81:       break;
 82:     case 'S':
 83:     case 's':
 84:       st = atoi(optarg);
 85:       break;
 86:     case 'E':
 87:     case 'e':
 88:       et = atoi(optarg);
 89:       break;
 90:     case 'h':
 91:     case 'H':
```

▶「編集室から」のコーナーの人々は，いくつもパソコンを所持している人が多いようですが，パソコン保有台数が増えるにつれ生活が圧迫されることはないのだろうか。ちなみに私は3台だけど，もう苦しくて，苦しくて。　小倉　圭司(24)東京都

```
 92:     case '?':
 93:       printf( "オプション説明：(nは整数です)¥n"
 94:           "  -in  CDROMのIDを指定します。デフォルトでは6。¥n"
 95:           "  環境変数CDROMにCDROMIDが指定されていた場合、それを利用しますが、¥n"
 96:           "  このオプションが指定されていれば、これが優先されます¥n"
 97:           "  -sn  スタートトラックの指定¥n"
 98:           "  -en  エンドトラックの指定¥n"
 99:           "  -p  Pause. CDの一時停止を行います。¥n"
100:           "  -r  Resume.CDの一時停止の解除を行います。¥n"
101:           "  -h  このヘルプをみます¥n");
102:       exit(-1);
103:       break;
104:     }
105:   }
106:   argc -= optind;
107:   argv = &argv[optind];
108: 
109:   init_inq();                          // SCSIユニットの初期化とINQチェック
110: 
111:   _scsi_ReadTOC_MSF( CDROM_ID,0x00,&SM,&SS,&SF,&min, &max );
112:                                        // 0x00は最始トラックを求めている
113:   _scsi_ReadTOC_MSF( CDROM_ID,0xaa,&EM,&ES,&EF,&min, &max );
114:                                        // 0xaaは最終トラックを求めている
115: 
116:   if(st==0) st=min;
117:   if(et==0) et=max;
118: 
119:   if ( st < 1 || st > max ) {
120:     printf("スタートトラックが不適当です¥n");
121:     exit(-1);
122:   }
123:   if ( et < 1 || et > max ) {
124:     printf("エンドトラックが不適当です¥n");
125:     exit(-1);
126:   }
127:   if ( st > et){
128:     printf("スタートトラックがエンドトラックよりもあとにあります¥n");
129:     exit(-1);
130:   }
131: 
132:   printf( "このCDは、最始曲=%dトラック  最終曲=%dトラック  です。¥n"
133:       "演奏時間は、%d分%d秒%dから%d分%d秒%dまでです。¥n"
134:       "%d曲目から%d曲目まで演奏します。¥n",min,max,SM,SS,SF,EM,ES,EF,st,et);
135: 
136:   _scsi_PlayAudioTrackIndex(CDROM_ID,(unsigned char)st,1,(unsigned char)et,1);
137: }
138: 
139: void  init_inq()
140: {
141: 
142:   while (_scsi_testunit(CDROM_ID) != 0) {
143:     printf("CDROM_ID:%d の装置は使用できません。¥n",CDROM_ID);
144:     exit(-1);
145:   }
146: 
147:   if(_scsi_inquiry(36,CDROM_ID,&s_inq) < 0){
148:     printf("CDROM_ID:%d の装置は使用できません。¥n",CDROM_ID);
149:     exit(-1);
150:   }
151:   if((s_inq.Qualifier_DeviceID & 0b00011111) != 5 ){
152:     printf("ID:%d の装置はCDROMデバイスではありません。¥n",CDROM_ID);
153:     exit(-1);
154:   }
155: }
156: 
157: //----------------------------------------------------------
158: //    以下は汎用ライブラリです。
159: //----------------------------------------------------------
160: 
161: void  _scsi_ReadTOC_MSF(
162:   int          ID,                     // SCSI ID
163:   unsigned char  track,                // 開始トラック
164:   unsigned char  *M,                   // M（戻り値としてのアドレス）
165:   unsigned char  *S,                   // S（戻り値としてのアドレス）
166:   unsigned char  *F,                   // F（戻り値としてのアドレス）
167:   int          *min,                   // 最始トラック（戻り値としてのアドレス）
168:   int          *max)                   // 最終トラック（戻り値としてのアドレス）
169: /*  説明：
170:   ReadTOC(43h)を実行しますが、戻り値を少なくするため、MSF形式に固定してあり
ます。
171:   また、手抜きで、トラックのコントロール属性を返していません。      */
172: 
173: {
174:   unsigned char  sendbuf[10];      // 送信データ
175:   unsigned char  receivebuf[12];   // ローカルワーク
176: 
177:   sendbuf[0] = 0x43;                   // scsi2 ReadTOC Command
178:   sendbuf[1] = 0x02;                   // LUN=0,Set MSF Bit
179:   sendbuf[2] = 0x00;                   // Reserved
180:   sendbuf[3] = 0x00;                   // Reserved
181:   sendbuf[4] = 0x00;                   // Reserved
182:   sendbuf[5] = 0x00;                   // Reserved
183:   sendbuf[6] = track;                   // 開始トラック
184:   sendbuf[7] = 0x00;
185:   sendbuf[8] = 0x0c;                   // アロケーション長
186:                                        // イニシエータが受け取ることの出来るバイト長
187:   sendbuf[9] = 0x00;                   // コントロールバイト
188:                                        // なにもしないので0。
189: 
190:   if( scsi_cmdout_ex( ID,10, sendbuf,12, receivebuf ) == 0 ) {
191:     *M   = receivebuf[9];              // M
192:     *S   = receivebuf[10];             // S
193:     *F   = receivebuf[11];             // F
194:     *min = receivebuf[2];              // Min. Track
195:     *max = receivebuf[3];              // Max. Track
196:   }
197:   else{
198:     printf(  "Read TOC(43h)がサポートされていません。¥n");
199:     exit(-1);
200:   }
201: }
202: 
203: int  _scsi_PlayAudioMSF(  // _scsi_PlayAudioMSFに従ってオーディオデータを再生する
204: //  引数
205:   int          id,                     // CDROMのID
206:   unsigned char  SM,                   // スタートM（分）
207:   unsigned char  SS,                   // スタートS（秒）
208:   unsigned char  SF,                   // スタートF（1/75秒）
209:   unsigned char  EM,                   // エンドM（分）
210:   unsigned char  ES,                   // エンドS（秒）
211:   unsigned char  EF)                   // エンドF（1/75秒）
212: /* _scsi_PlayAudioMSF(47h)を実行して、CDROMからオーディオデータを再生します。
213:   データはunsigned char形式で、分、秒、1/75秒で設定せねばなりません。  */
214: 
215: {
216:   unsigned char  sendbuf[256];
217: 
218:   sendbuf[0] = 0x47;                   //  Play Audio MSF Command
219:   sendbuf[1] = 0x00;                   // LUN=0,Reserved
220:   sendbuf[2] = 0x00;                   // Reserved
221:   sendbuf[3] = SM;                     // 開始：M
222:   sendbuf[4] = SS;                     // 開始：S
223:   sendbuf[5] = SF;                     // 開始：F
224:   sendbuf[6] = EM;                     // 終了：M
225:   sendbuf[7] = ES;                     // 終了：S
226:   sendbuf[8] = EF;                     // 終了：F
227:   sendbuf[9] = 0x00;                   // コントロールバイト
228:   return( scsi_cmdout_ex( id,10, sendbuf, 0, sendbuf ) );
229: }
230: 
231: int _scsi_PauseResume( int id ,unsigned char PR)
232: /*  説明
233:   CDROMをPauseしたりResumeしたりする。
234:   PR  0  ポーズ  べつにCDROMがナイスポージングを決めるわけではない
235:   PR  1  レジューム
236: */
237: 
238: {
239:   unsigned char  sendbuf[256];
240:   sendbuf[0] = 0x4B;                   //  Play Audio MSF Command
241:   sendbuf[1] = 0x00;                   // LUN=0,Reserved
242:   sendbuf[2] = 0x00;                   // Reserved
243:   sendbuf[3] = 0x00;                   // Reserved
244:   sendbuf[4] = 0x00;                   // Reserved
245:   sendbuf[5] = 0x00;                   // Reserved
246:   sendbuf[6] = 0x00;                   // Reserved
247:   sendbuf[7] = 0x00;                   // Reserved
248:   sendbuf[8] = PR;                     // Pause & Resume
249:   sendbuf[9] = 0x00;                   // コントロールバイト
250:   return( scsi_cmdout_ex( id,10, sendbuf, 0, sendbuf ) );
251: }
252: 
253: 
254: int _scsi_PlayAudioTrackIndex(
255:   int id,                              // ID
256:   unsigned char STrack,                // スタートトラック
257:   unsigned char SIndex,                // スタートインデックス
258:   unsigned char ETrack,                // エンドトラック
259:   unsigned char EIndex)                // エンドインデックス
260: /*
261:   スタートトラック、インデックスとエンドトラック、インデックスを指定して、
262:   オーディオデータを再生する。
263:   再生は、PlayAudioTrackIndex(48h)をコールして実現している。
264: */
265: {
266:   unsigned char  sendbuf[256];
267: 
268:   sendbuf[0] = 0x48;                   //  Play Audio MSF Command
269:   sendbuf[1] = 0x00;                   // LUN=0,Reserved
270:   sendbuf[2] = 0x00;                   // Reserved
271:   sendbuf[3] = 0x00;                   // Reserved
272:   sendbuf[4] = STrack;                 // Reserved
273:   sendbuf[5] = SIndex;                 // Reserved
274:   sendbuf[6] = 0x00;                   // Reserved
275:   sendbuf[7] = ETrack;                 // Reserved
276:   sendbuf[8] = EIndex;                 // Reserved
277:   sendbuf[9] = 0x00;   '               // コントロールバイト
278:   return( scsi_cmdout_ex( id,10, sendbuf, 0, sendbuf ) );
279: }
280: 
281: int  scsi_cmdout_ex(
282: //  引数
283:     int   id,                          // SCSI ID
284:     int   len,                         // 送信データ長
285:   const void  *sendbuf,                // 送信データ配列
286:     int   rlen,                        // 受信データ長
287:     void  *receivebuf                  // 受信データ配列
288: )
289: /*
290:   説明：
291:   CDB形式のコマンドをIDで指定されたターゲットに転送します。
292:   CDB形式コマンドは、sendbufで示された番地から、配列変数でlenの長さだけ定義
293:   してください。単に送るだけではなく、戻り値を取り、receivebufに受信データを
294:   もどします。受信データ長は、十分な長さがあらかじめ必要です。0であった場
合、
295:   ステータスインフェーズ、データインフェーズを実行しません。
296:   この命令を実行するには、SCSIドライバが常駐している必要があり、インクルード
297:   ファイルとしてsys/scsi.hが読み込まれている必要があります。      */
298: 
299: {
300:   unsigned char status,msg;
301:   int req;
302: 
303:   if( (req=_scsi_select(id)) == 0) { // アービトレーション&セレクションフェーズの
発行
304:     if( (req=_scsi_cmdout( len, sendbuf )) == 0) { // コマンドアウト
305:       if( rlen != 0 ) {
306:          if( (req=_scsi_datain_p( rlen, receivebuf ))==0) {
307:           if( (req=_scsi_stsin( &status )) == 0 ) {       // ステータスインフェーズ
308:             if( (req=_scsi_msgin( &msg )) == 0) {         // メッセージインフェーズ
309:               req = (msg * 256) + status;
310:               return(req);
311:             }
312:             else {
313:               printf("メッセージインフェーズ中にエラーが発生しました¥n");
314:               return( -1 );
315:             }
316:           }
317:           else {
318:             printf("ステータスインフェーズ中にエラーが発生しました¥n");
319:             return( -2 );
320:           }
321:         }
322:         else {
323:           printf("データインフェーズ中にエラーが発生しました。エラーコード=%d¥n
",req );
324:           if(req < 0 ) return(req);
325:           if(( req = _scsi_stsin( &status ))==0){
326:             if(( req = _scsi_msgin( &msg ))==0) {
327:               req = (msg * 256) + status;
328:               return(req);
329:             }
330:             else {
331:               printf("メッセージインフェーズ中にエラーが発生しました¥n");
332:               return( -1 );
333:             }
334:           }
335:           else {
336:             printf("ステータスインフェーズ中にエラーが発生しました¥n");
337:             return( -2 );
338:           }
339:         }
340:       }
341:       else {      // 受信がないとき
342:         if((req = _scsi_stsin( &status ))==0) {
343:           if((req = _scsi_msgin( &msg ))==0) {
344:             req = (msg * 256) + status;
345:             return(req);
346:           }
347:           else {
348:             printf("elserlen:メッセージインフェーズ中にエラーが発生しました¥n");
349:             return( -1 );
350:           }
351:         }
352:         else {
353:           printf("elserlen:ステータスインフェーズ中にエラーが発生しました¥n");
354:           return( -2 );
355:         }
356:       }
357:     }
358:     else {
359:       printf("コマンドアウト中にエラーが発生しました¥n");
360:       return( -3 );
361:     }
362:   }
363:   else {
364:     printf("アービトレーション&セレクトフェーズ中にエラーが発生しました¥n");
365:     return( -4 );
366:   }
367: };
```

▶仕事でBASICを使うようになり，少しはBASICがわかってきた気がします。
長谷川　常吉(34)三重県

X680x0周りのSCSIを探る!

Taki Yasushi 瀧 康史

今月は工作からちょっと離れて周辺機器のつなぎ方などについて考えてみます。ビデオユニットをテーマに，SCSIの基本を探ってみましょう。あなたの接続法は大丈夫ですか?

哀愁のイメージユニット2

CZ-6VT1を持っている人は,その気になって探せば割といます。私も昔は持っていました。だけどCZ-6VS1を持っている人となると，そうはいません。私の周りではほんの2，3人。これ，今年度唯一の新製品(ハードウェアの)なんだけどなあ……なんて思ったりして。

CZ-6VT1,通称イメージユニットといわれるものです。X680x0にあるIMAGE IN端子を利用して,高速にテレビ画像をVRAMに展開するハードウェアです。時代ものなので画質が悪いとか，値段が7万円弱で割高だとか，いろいろ欠点はありますが，かなりユーザーが多いのも事実です。

そして，CZ-6VS1。去年(1994年)の春，シャープが新Xも出さずに唯一出したハードウェアです。データの転送手段などは違っても，実はイメージユニットと目的をほぼ同じくするもので，通称イメージユニット2といわれています。かつて荻窪氏が切に登場を待っていた様子ですが(本誌バックナンバーより推察)，定価178,000円というゴージャスなお値段。彼ははたして購入したのでしょうか?

なお，CZ-6VS1は，私に今回の記事を書かせた「憎い奴」，もとい「愛い奴」で今回の主人公です(なんだそりゃ)。

さて私は最近，画像データの取り込みに凝っています。どんなの?　といわれると，困ってしまいますが，なんとなく「取り込む媒体」よりも，「取り込む」という事実のほうへ関心がいってしまっているんですね。あぁ，これで私もマルチメディア!　なんてバカバカしいこというつもりじゃないですけど，なんとなく安いVHSテープに3倍で録画を何度も繰り返したくらい画質が落ちたテレビ画像を,ウインドウ内(ライブスキャン)で見てみると,マルチメディアって感じがしますよね。私だけかもしれないけれど。

それで，このライブスキャンというのが，CZ-6VS1用に作られた取り込みツールです。なんとSX-WINDOWの片隅でテレビ放送などが見られます。とはいえ，まともに見るためには160×120(それでも6万色)程度のサイズになってしまうんですけどね。768×512サイズのモニタならこれでもよいのですが，1024×1024サイズのモニタだと，もはやなにが映っているのかわかりません。

おまけに，これをやると慢性的に強烈な重さを体感できるので，私は速いマシンを使っている恩恵を思い出すためにときどきライブスキャンを立ち上げます。これはもうX68000-8MHzでSX-WINDOWを使っている感じです。はっきりいって，普通にテレビを見るためならば，私だって素直に隣にあるブラウン管を使います。おんなじブラウン管でテレビを見ようとは思いません。

ライブスキャンのことはともかく，取り込みのほうに話を戻しましょう。

ことの発端は友人がCZ-6VT1で取り込んだ画像データを多量に持ってきたことです。全部アニメものですが，私にとっては本当に多量。見たことないものばっかだし，MO9枚を数えています。

SX-WINDOWベースで仕事を行う私は，ガシガシSX-WINDOW標準のCGA形式に変換してGRW内で見る始末。しかし,CZ-6VT1で動画を取り込むと汚いのなんの。話によると，取り込み速度，128×128ぐらいで15枚/秒ぐらいでしょうか?　いって20～30枚/秒。取り込み速度は申し分ないのですが，メモリ取り込みなので，アニメのオープニング1本とるのに，結構苦戦を強いられるとのこと。とにかく粗いし，汚いのです。

そこでCZ-6VS1を使えばどこまでいけるかなあ?　という疑問が湧き，急遽システムにCZ-6VS1を導入してみることにしました。

注) 記事中にはSCSIがらみの内容がたくさん出てきます。特に記述がない限り，SCSIはSCSI1,SCSI2を含めてを意味し，SCSI1はSCSI2から見ての旧SCSIを意味します。

また，SCSIと一般にいう場合，現在においてはSCSI2レベルであることを明示しておきます。

CZ-6VS1の設置

とにかく白いボディを設置。ああ，黒の統一美を崩してしまうけど，もはや，我が家の黒いボディはX68030とCZ-6EB1・BK(拡張I/Oボックス)とSC-55と以前塗ったCM-64だけです。ほかのものはすべてリニューアルされて,みんなベージュ(オヤジカラー)になってしまっているのですね。いっそのこと,X68030をオヤジカラーで塗ってしまおうか?とも思いましたが,「だせぇ」という天の声に救われました。

はてさて，CZ-6VS1でライブスキャンを使ってなにかを録画します。本来なら320×240ドット/6万色ぐらいで攻めたいところですが，ライブスキャンでリアルタイムに取り込み,テレビを見るという状態(すなわち保存せずに垂れ流すモード)だと,もはやX68030 36MHz改でも，上から下へ書いているのがわかるほど遅くなってしまいます。どうやら160×120ぐらいで妥協せねばならないようです。

もっとも，320×240の解像度でX680x0シ

リーズでまともにアニメーションさせるためには大変な努力がいると思いますが(いわゆる，X680x0シリーズの最大の弱点といわれているものです)。これがPC-98GSのように，台形転送機能[*1]があったり，どこぞのマシンのように，自由形写像転送機能[*2]があったら凄かったんですけどね。

なにはともあれ，サイズは160×120で諦めたほうがよいようです。「これではCZ-6VT1とあんまり変わらないじゃないか！」といわれそうですが大正解です。動画をやるという目的で導入したのに，これではなんの意味もありません。

まあ，とりあえずはエンドユーザーとして使ってみることにします。1枚絵の取り込み(スチルモード)の美しさは，私としても大満足（CZ-6VT1よりもずっと綺麗です)。これだけでも，178,000円の価値はあると考えることにしました。

CZ-6VS1の取り込みモードは3種あります。まず，

1) X680x0本体のメモリへ取り込むモード
2) X680x0にHuman68kによって接続されたHDDへ取り込むモード
3) CZ-6VS1専用のHDDを確保して，それにテンポラリとして書き込むモード

この3つです。

図1では便宜上，左から右へデータが流れるようになっていますが，実際の接続はSCSIケーブルはどのような順序でも原理的には変わらないはずです(図2のように)。

ここに私もはまった落とし穴があります。この時点で，私はどう考えても1)がいちばん速く，ついで3)，2)の順になると考えていました。その根拠は簡単で，まず1)のメモリモードはHDDをアクセスしなくてよいのですから当然いちばん速いに違いないというのがその理由。次に3)と2)の違いとして，2)は一度X680x0を通ってDOSとしてのHuman68kを通して保存するのに対して，3)は直接CZ-6VS1がHDDに保存するわけです。だから3)のほうが速いに違いない，と考えました。

そこで160×120を(1)の方法で，メモリに録画してみたのですが，結果は10枚/秒。アニメーションならあまり問題はないかな？って感じです。画面がパンニングするときに秒間10枚の悲しさが出てきますが，それでも画面は綺麗なので，CZ-6VT1よりはマシでしょうか？　しかし，ある意味，動画のことは全然考えていないCZ-6VT1でさえ15枚/秒が実現できたのに，CZ-6VS1って単に綺麗なだけかとも思ってしまいました。くそぉ。

ともあれ，買ってしまったものはしかたがないのですから，有効利用を考えます。このままでは本当に綺麗なだけです。ここで諦めるのも悲しいので，2)のHDDに録画するモードを利用することにしました。これがうまくいけば，簡単に多量の画像を取れる分だけでも，CZ-6VS1の価値はあるでしょう。

ところが(2)番の方式でHDD録画したところ，CONNERの545Mバイトドライブへ直接録画して最大8枚/秒程度。決して遅いHDDではありません。しかし一気に90秒ぐらい簡単に録画できる(HDDの残り容量に比例する)ので価値はありますけど，やっぱり秒間8枚ではカクカクもいいところです。

くそぉ。今回二度も苦汁を味わってしまいました。

図1　CZ-6VS1の3つの取り込みモード

*1　台形転送機能は平たくいえば，ある台形のグラフィックを，別の部分にハードウェア的に一瞬にして転送する機能です。どんなに大きくてもウィンドウの表示が一瞬にしてでき，ウィンドウのドラッグをNeXTみたいにずりずり引きずっても重くならないという，あれば最高な機能なのですが，残念ながらX680x0にはついていません。ついていればかなり凄いパソコンだったはずなのですが。

余談ですが，私はPC-98GSが好きで1台すでにキープしています。疑似スプライト（TOWNSのそれよりもずっと強力であるし，おそらくX680x0のスプライトよりも結果的に強力でしょう）もついてるし，画面機能はX68000よりも上だと思っています。ただ，資料さえ公開してくれていたら……あることしかわからない機能なんてないも同じです。

*2　自由形写像転送機能というのは，いわゆる台形転送機能がさらに強化され，このサイズが任意に変えられるというものです。つまり2次元イメージグラフィックを単に転送するだけでなく，自由形に座標変換しながら転送できるものです。これさえあれば3次回転しながらうりうり動かしたり，拡大したり縮小したりできるというわけ。もちろんテクスチャマッピングだってできます。これがX680x0についていたら，ウィンドウがくるくる回転しながら開く！　とかできたんですけれども。きっと次世代ゲーム機といわれているマシンには，これに近い機能がついているのでしょう。

イニシエータとしてのCZ-6VS1

こうしてCZ-6VS1は忘れられるのでしょうが，やっぱりね。ちょっとやそっと，使っただけで，「ダメ」と結論づけてしまうのはまだ早すぎます。

メモリ録画で10枚/秒。HDD録画で8枚/秒いったのだから，直接HDDに録画すれば9枚/秒ぐらいはいくはず，うまく10枚/秒いけば，連続で90秒ぐらい簡単に綺麗に取れるCZ-6VS1はそれでも価値があるではないですか！

以下，3)のHDDに直接録画するモードのことを，高速録画モードといいましょう（ライブスキャンでそう記述されているので）。

そこで秋葉原にHDDを購入しに行きました。このHDDはCZ-6VS1に管理されるHDDです。接続は図1のようにX680x0本体までディジーチェーンされますが（おそらく）高速録画モード時にはCZ-6VS1がイニシエータになって，QUANTUM 270Mバイトドライブを動かすはずです。この時点でのSCSIデバイス一覧はこうなっていました。

ID0：CONNER　CP30540 545MB3.5B1C2
ID1：CONNER　CP30540 545MB3.5B0BE
ID2：QUANTUM　LPS270S　5906
ID4：TOSHIBA　OD-D300　000B
ID5：*SHARP*　CZ-6VS1　1.00
ID6：SONY　CD-ROM CDU-55S1.0f
ID7：SHARP　X68030

新規購入したQUANTUM 270Mバイトドライブは，TEXAのHDDユニットの中身[*3]です。インタフェイス込みで26,000円。QUANTUMドライブは速い速いと聞いてはいたのですが，使ってみてびっくり。激速ではないですか。とりあえず，普通に使ってみるためにフォーマットして，DSK BENCH ver.0.21（©じゃぎゅあ）でチェックしたところ，Rnd.read 64Kバイト時に1200Kバイト/秒いきます。同DSKBENCHをCONNER CP30540にかけたところ，こちらも1200Kバイト/秒でした（当然シーケンシャルではもっといきます）。いずれもHSCSIというX68030のSCSI速度を高速化させるソフトを利用して，さらにバリバリなクロックアップをしてますが。

話が脱線しましたね。とにかくCZ-6VS1に利用するHDDは，フォーマットのみして，領域確保をしていないドライブと決まっています。要するにCZ-6VS1用のHDDは，テンポラリ以外には利用されないということです。さすがの私もいささかもったいない気はします。確かに。

とにかく，単純に考えると，取り込んだ画像を直接落とすドライブなのですから，速いHDDがよいに決まっています。大きさは録画する時間によりけりなので，適当に選んでよいとは思いますが，まあ，普通270Mバイトもあれば十分でしょう。

ほかにこんなことをする人がいるのだろうか？　と思いつつ，買ってきたHDDの領域を解放した状態にします。ひょっとしたら左のSCSIデバイス一覧で気がついた方がいるかもしれませんが，秘かにID7：SHARP X68030以外はすべてSCSI2デバイスです。当然すべてsyncronous（同期転送）モードを装備しています。CZ-6VS1の広告にも載っているとおり，高速録画モードで利用するHDDはSCSI2で同期転送モードをサポートしているものと決まっています。

図2　CZ-6VS1導入時の我が家のSCSI状況

まずはなにも考えず，ライブスキャンを実行します。SHIFT＋録画設定で高速録画モードに設定できるなんて，マニュアルをきちんと見るまで気がつきませんでした。

設定を終えて録画ボタンを押してみます。すると，CZ-6VS1のBUSYランプも，QUANTUM HDDのBUSYランプもどちらも消灯して，SX-WINDOWがハングアップしてしまいました。

くそぉ。なぜでしょう？　今回3度目の，「くそぉ」です……。

＊3　なにもHDDに限った話ではないのですが，ドライブまで作ってしまえるほど大きなメーカー以外，一般に中身のドライブは自社製ではありません。この場合，TEXAというメーカーのHDDの中身をのぞいてみたら，QUANTUMのドライブを（部品として？）使っていたという話です。

高速モードが使えた！

とりあえず，ショップに購入したばかりのQUANTUMドライブを持っていって，事情を話し，ちゃんとCZ-6VS1で動くドライブはないか探すのを手伝ってもらいました。ところがショップで用意できたSCSI2ドライブはすべて全滅。INQUIRY[*4]をとってみると，型版は違ってもすべてQUANTUM製。どうやら試してみたQUANTUM製のドライブはすべてCZ-6VS1では全滅した様子です。広告にはFAST SCSI2対応のデバイス[*5]と書いてあるため，INQUIRYでSCSI2対応，同期転送モードがサポートしていると出てくるドライブでしか使えないはずです。QUANTUM製のドライブは日本ではかなりシェアを取っているドライブのはずですから，CZ-6VS1はいったいどんなドライブで実験したのでしょうか？

残念ながらFAST SCSI2対応ドライブのものはほかにショップにないため，たまたまあったSCSI1の同期転送モード非サポートのドライブ（TOSHIBA MKシリーズ540Mバイトモデル/緑電子のNOVA2に搭載のドライブです）でも試してみることにしました。このドライブは，2.5インチHDDのため速度が結構速いのですが，インタフェイスは，SCSI1で，さらに同期転送モードは持っていません（つまりCZ-6VS1非対応のドライブです）。

ところがSCSI1のドライブなのに，うま

くいくじゃないですか！ 160×120で26枚/秒。メモリ転送で10枚/秒なのにどうして？ と一瞬目を疑ってしまいましたが，何度やっても26枚/秒前後転送します。結局，たまたま中古で在庫のあった200MバイトのNOVA2でもうまくいくだろうということで，これを一応買ってくることにしました。約２万円の出費です……*6。

それにしても，SCSI2ドライブではうまくいかず，唯一うまくいったドライブが，SCSI1だなんて……。結局，最初に買ったQUANTUM LPS270Sの利用方法を考えつつ，その日は帰宅しました。

*4 INQUIRY SCSIで定められた命令のひとつで，平たくいえば，SCSI機器に対して「あなたはどういう人ですか？」と聞く命令です。これによって，そのデバイスがHDDであるとか，MOであるとか，CD-ROMであるとかがわかります。それだけではなく，オプショナルの通信モード(たとえば同期転送モードなど)のサポートや，製作メーカー(ベンダー)や品名などもわかるというものです。
*5 FAST SCSI2 SCSI2のモードで高速な転送レートを持ったものです。後ろの項で詳しく説明しますから，ここではとりあえずSCSI2で同期転送モードサポートのものは，ほぼFAST SCSI2対応と思っていてください。
*6 げー，よく買うなあ。と思われるかもしれません。本文中では１日で終わったように見えますが，実際は３日ほど，毎日，閉店ギリギリまで，お店の人につきあってもらっていたという手前，これ以上迷惑をかけるわけにはいかないという配慮があったからだったんですが……。それにしてもCZ-6VS1，憎い奴，もとい，愛い奴。

8つのIDで行き詰まる

とにかくCZ-6VS1で使えるドライブが我が家に入ってきました。が，適当にサクサク繋げてみると，起動さえしません。IDはあってるのにおかしい。

このときの様子は，

ID0：CONNER　CP30540545MB3.5B1C2
ID1：CONNER　CP30540545MB3.5B0BE
ID2：QUANTUM　LPS270S　5906
ID3：TOSHIBA　MK2224FB　920C
ID4：TOSHIBA　OD-D300　000B
ID5：SHARP　CZ-6VS1　1.00
ID6：SONY　CD-ROM CDU-55S1.0f
ID7：SHARP　X68030

なんと，全部ID埋まってるではないですか！ こんな環境，あんまりないだろうなぁ……とか思いつつ，いろいろ考えてみました。

接続としては，図２のSHARP CZ-6VS1とSONY CDU-55Sの間にTOSHIBA MK2224FBを入れました。家の結線としては，TOSHIBA OD-D300以外は，ハーフピッチなので，フルハーフのケーブルをひとつにするにはOD-D300は最後でなくてはいけません。

とにかく，SCSIというのはケーブル長などまで影響する(というのをマニュアルで読んで知っていた)そうなので，デバイスを少しずつ外してみました。結果，末端のOD-D300を外しただけで動き，面白いことに途中のCD-ROMだけを外してもやはり動くのです。当然，いままで使っていた状態ではTOSHIBA MK2224FBがなかったのだから，これがなくても動くということになります。

つまりはシングルエンド非平衡型の最大総ケーブル長である6mを超えたってことになるのでしょうか？ どう考えても6mはなさそうなので，この6mはきっと，環境によって変わってくるのでしょう。

とりあえずはOD-D300を外せば起動もフォーマットもできるようなのでしばらくこれで使うことにしました。そしてSX-WINDOWを起動してライブスキャンを起動します。高速録画モードにして，録画開始……うーん動かない。あれぇ？

そこで，なぜかたくさん持っているターミネータを代わりばんこに，適当につけてみると，なぜか１個だけ動くターミネータがありました。これで録画してみたら160×120で16枚/秒取れるようです。ときどきHDDがハングアップするなど不安定でしたが，ここまで結構苦労したのでうれしさもひとしおです。ショップで26枚/秒いったのは，あちらが500Mバイトのモデルだからでしょう。普通，HDDは大きいほうが速いのですから。

大きいドライブのほうが速いのならと，手持ちのHDDのなかで実験していなかったCONNERドライブを試してみることにしました。

余計なSCSIデバイスを外し，以下の状態でチェックを行いました。

ID0：CONNER　CP30540 545MB3.5B1C2
ID1：CONNER　CP30540 545MB3.5B0BE
ID5：SHARP　CZ-6VS1　1.00
ID7：SHARP　X68030

接続は図３の通りです。

一応断っておきますが，CONNER CP30540というドライブはFAST SCSI2の同期モードありのドライブです。結構高速なHDDです。

この状態でライブスキャンを起動し，高速録画モードにしてみました。

「動く！」

なんと，160×120/６万色モードで30枚/秒も取れます。テレビのスキャンから考えると30枚というのはリミットなのでしょう。４倍のサイズの320×240/６万色モードで10枚/秒。再生ソフトさえなんとかすれば，このサイズでも10枚というのは，なかなか魅力的な数字です。

面白いと思って，さらに30秒ほど録画したら今度は止まってしまいました。もの凄く不安定なようです。二度も成功したということは，このドライブでちゃんと高速録画モードが使え，取り込みができたということです。ケーブルをいろいろ取り替えてみたところ，それで安定度が変わることがわかりました。

また，我が家のX68030は，多少強引な方法で内蔵HDDが接続してあります。初めケーブルはCZ-6VS1とCZ-6VS1用のテンポラリHDDの間だけ良質なものを使っていましたが，よくよく考えるとSCSI信号は高周波になります。高周波になると，無意味

図３　CONNER DRIVEで実験をしたときのSCSI状況

に結線されている線がバカにできなくなるものです。内蔵HDDはフラットケーブルで，なにも考えずとりつけていましたから，この場で内蔵HDDを取り外してみました。

結果は良好。わざと図1，2では内蔵HDDを本体よりも左に書きましたが，よく考えてみれば，この図中，X68030の左側にターミネータがあるべきです。ターミネータはあくまでも終端につけるものののはずですが，それが中間についているのですから，安定するはずはありません。

ここでようやく，一段落ついた感じがし，CZ-6VS1をやっとフルに使いこなせた気もしてきました。

SCSI転送の話

日記はこれで終わりです。これでやれやれと終わってしまうのなら，私はOh!Xのライターをする資格はないでしょう。これだけ大変な思いをして，これだけ金をかけたのだから，同じことをこれからしようと思っている人のためにもきちんと記事にしておかねばなりません。

これまで，

「SCSIは難しい」

といういろいろな友人の暗示が頭から離れなかったために，どうしてもSCSIについて，真面目に考える気が起きませんでした。できるのならSCSIがらみについては，ずっと使うだけのエンドユーザーでいたい。なんて甘いことを考えていたのですね。だけど，わざわざ専用HDDまでも購入してまでして，ここであとにひけるはずはありません。とりあえず，それを心に秘めつつSCSIの勉強を始めることにしました。

ここで，現在の状態までの出来事を考察することにします。

まずはなぜ，高速転送モードはX68030-36MHzマシンのメモリに転送するよりも，HDDという機械装置に転送したほうが速かったのでしょうか？　最初に予測した3つのモードの速度は，見事に外れています。通常，半導体内でのやりとりよりも，機械となるHDDのほうが遅いのですが，ここではそれを裏返すような結果が出ました。

この理由はわりと簡単にわかりました。疑うべきはデータの転送速度しかないからです。HDDは機械制御のものでありながら，非常に高速なものだからです。いったいX680x0の転送速度がどのくらいなのか，とある資料の片隅を見たら，マシンを問わず(X68030でも)だいたい900Kバイト/秒と書いてありました。

SCSIのデータ転送は，基本的には非同期で転送されます。その具体的な方法は，データを「送る側」がまずREQ信号(リクエスト)を「もらう側」に送り，「もらう側」はREQ信号を受けたという確認として，「送る側」にACK信号(アクノリッジ)を戻します。信号がこうして1往復する間に，「送る側」は8ビットのデータバスにデータを乗せておくのです。「送る側」がACK信号(アクノリッジ)を受け取ったら，再び同じ処理を繰り返します。言葉ではわかりづらいでしょうから，図4を見てください。

非同期転送は，「送る側」「もらう側」の反応速度が同じでなくても，データの転送ができる利点がある半面，このように，たった1バイトのデータを送るために，2つの信号が往復します。

図4でわかるとおり，このデータ転送速度は「もらう側」と「送る側」それぞれの応答速度に影響しますし，ケーブルの中を信号が伝送される時間も影響します。つまりは，ケーブル長にも影響するということです。

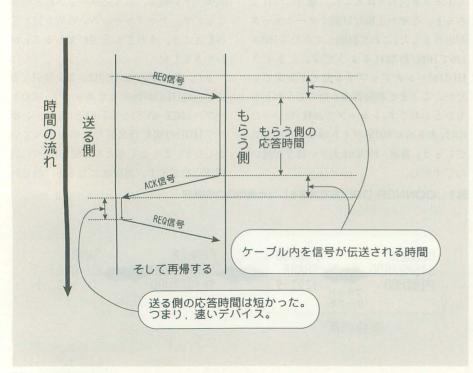
結果，このデータ転送方法では速くても3Mバイト/秒程度しかデータ転送できないというわけなのです。資料によれば，実質的に非同期転送は2M～3Mバイト/秒*7で送られているそうですから，X680x0の転送レートは非同期にしても「かなり」遅いほうです*8。

3Mバイト/sec非同期転送できれば，かなり速いと私は思うのですが，これ以上速度を上げなければならないケースもままあるでしょう。たとえば，画像などはよい例です。そこでSCSIではデータの転送を高速にするために，データ転送を同期転送するモードを(オプションで)持っています。

この同期転送というのは，ACK信号とREQ信号をパルス検出して一気に転送を行う方法です。具体的には図5を見ていただければわかると思います。

まず，データを転送する速度をあらかじめ2つのデバイス間で決定します。転送速度は，SDTR(Synchronous Data Transfer Request)メッセージを2つのデバイスの間で送りあうことによって決まります。

この転送速度の最大値はSCSI1規格では5Mバイト/秒までで，SCSI2では10Mバイト/秒となっています。一般にFAST SCSI2というのは，このSDTRの値が5Mバイト/秒を超えた同期転送モードのことをいいます。SCSI1で同期転送がサポートされてい

図4　非同期転送時の様子

たドライブというのを探すのは難しく，私が知っている限りではSONYのMOドライブぐらいのものです。これとて，SCSIオプションとしての同期転送を確実に使っているかは謎です。

さらに，SCSI2 HDDはIDE HDDが高速になってきたという触発もあってか，高速同期モードをサポートしている場合が多いので，SCSI2 HDDなら，ほぼ5Mバイト/秒以上の同期転送に対応しているようです。結局，同期転送をサポートしているドライブ＝SCSI2という式がほぼ成り立っています。

転送速度の話はこれで切り上げて，転送の手順について説明をしましょう。ここで，送る側はターゲット(この場合HDD)で，もらう側はイニシエータ(この場合CZ-6VS1)とします。

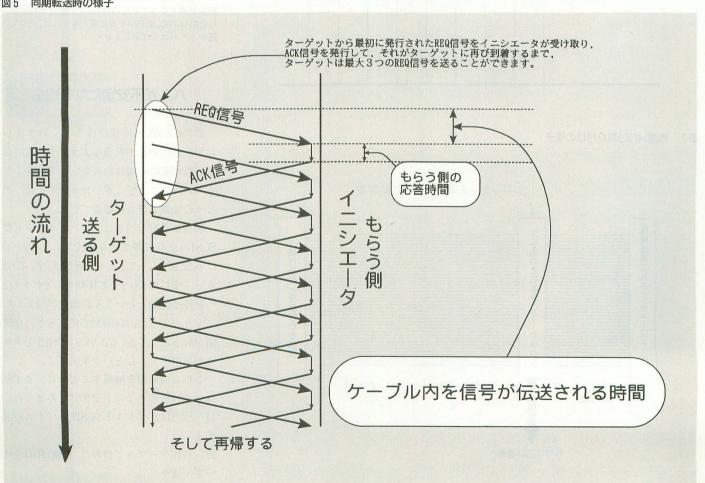
図5でわかるとおり，まずターゲットはREQ信号を発行します。最初に発行されたREQ信号をイニシエータが受け取り，ACK信号を発行します。これだけでは非同期転送と同じですが，同期転送の場合，送る側，すなわちターゲットは，ACK信号の到着を待たずに，あらかじめSDTRで決められた時間(転送速度の逆数。5Mバイト/秒なら100ns)になったら，次のREQ信号を発行するのです。実際のデータはデータバスにREQ信号と同じタイミングで乗せられます。

こうすることにより，転送制御の応答時間は最初のACK信号が返ってくるまでの時間を除き，SDTRで定められた時間になります。データ転送の辻褄をあわせるには，REQ信号とACK信号のパルスの数を確認します。具体的には，ターゲットは発行したREQ信号に対するACK信号が返るまでに，いくつ自分がREQ信号を発行するかを数えておき，イニシエータは発行したACK信号がREQ信号になって返ってくるまでに，いくつACK信号を発行したかを数えておき，この2つが等しいことをチェックするのです。規格では最大で，ひとつの信号がいって返ってくるまでに3つのREQ信号を発行するのが許されています。

CZ-6VS1の転送レートは，2.8Mバイト/秒だそうです。広告にはFAST SCSI2対応ドライブのみ，テンポラリHDDとして対応と書かれていますが，同期転送の速度は「FAST」という冠詞がつくほど高速ではないようです。まあ，SCSI2ドライブならば，普通はCZ-6VS1対応と考えてよいのかもしれません。2.8Mバイト/秒という中途半端なスピードは，きっと，CZ-6VS1が内部でデータを取り込む最小時間からきているのでしょう。

勘のよい方はもうわかったでしょう。CZ-6VS1がX680x0のメモリにデータ転送するためには，X680x0にあわせて900Kバイト/秒以下までに落ちてしまうということです。さしずめ，図4では送る側がCZ-6VS1，もらう側がX680x0といったところでしょうか。

もっと具体的に説明しましょう。図1のSCSI転送(1)の部分は，900Kバイト/秒(非同期なので実際はそれ以下)です。それに対

図5　同期転送時の様子

して(3)のSCSI2転送は2.8Mバイト/秒(同期)です。もちろん，同期モードを利用しているでしょうから，環境によって前後することはありません。だから，データ転送は当然高速になります。最近のHDDの転送速度は900Kバイト/秒では賄えないほど高速ですから，(3)の高速録画モードは本当に高速になったということなのです。

参考までに高速転送モードが使えたHDDと使えなかったHDDを列挙しましょう。

FUJITSU　M2614S　(SCSI1)
TOSHIBA　MK2224FB　(SCSI1)
TOSHIBA　MK2326FB　(SCSI1)
CONNER　CP30540 545MB3.5　(FAST SCSI2)

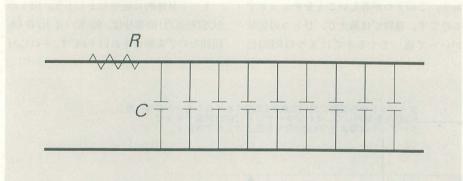
結果的に，この4つは確認ができました。CONNERドライブを除いて，上の3つのHDDは，INQUIRYチェックをすると，SCSI1規格のみ準拠で，同期転送オプションは使えません。つまり非同期転送のみのデバイスです。

CZ-6VS1のマニュアル中では，FAST SCSI2のみ対応と書かれていますが，実際には，CZ-6VS1内部でデバイスをチェックして，SCSI1デバイスでも録画できるようにしてあるのでしょう。

ただし，使えるといっても，やはり速いディスクでないと，高速な転送はできないらしく，秒間のコマ数はかなり落ちてしまいます。おすすめは，CONNER CP30540でしょうか。LogitecのManhattan LHD-B540，ELECOMのFixell 540内蔵のHDDでした。普通にHuman68kでフォーマットして使っても結構速いドライブです。もっとも，このドライブは030以前のマシンでは，起動が多少不安定なようでしたが……。

ところで，なぜQUANTUM LPS270SはFAST SCSI2対応なのに，使えなかったのでしょう？　このハードディスクはTEXAのStation Pro260に内蔵のHDDで，普通にX680x0シリーズにつなげると，文句ない高速さでカリカリ読み書きしてくれます。X680x0とは相性ばっちりのようですが，CZ-6VS1とは相性はいまいち悪いようです。

TEXAのインタフェイス(PC-9801用)のマニュアルを見ると，インタフェイス側は同期転送モードを持っていて，このHDDと最大10Mバイト/秒で転送できるそうですから，きちっと同期転送がサポートされているドライブのようです。

このドライブがなぜ接続できなかったのかは，どうしても解明できませんでした。

＊7　2Mバイト〜3Mバイト/秒とはいっても，非同期転送できっちり，2Mバイト/秒をマークしているSCSI機器は少ないと思います。根拠はありませんが。

＊8　X68000 SUPERを作ったときと同じアーキテクチャだからでしょう。互換機やPC-98地方でも，SCSIが速くなってきたのはSCSI2が規定されてからの話なので，SUPERが出た頃で考えれば，決して遅かったわけではありません。

図6　高周波における並行二線

図7　内蔵HDD取り付けの様子

バスが不安定になる理由

結局CZ-6VS1の接続はうまくいきましたが，なにかと不安定なようです。現時点での我が家の環境は割合安定しているようですが，ここまでこぎつけるまでには，かなりの知識と努力が必要でした。

最初に考察すべきは，シングルエンド型SCSIの全長限界である6mを超えていないのに不安定になってしまった状態です。IDは0〜7まで埋まっている状態で，どれかひとつを外せば動くという不思議な状態でした。

もうひとつは，X68030で使うときには問題が起きなくても，CZ-6VS1で使うとエラーが出やすかったという事実です。

これらの原因を解明するために，まず私はいくつかのポイントを挙げてみました。

1)　SCSI2デバイスとSCSI1デバイスの混在
2)　自作ケーブルで留めた，内蔵HDDのケーブル周り
3)　ケーブルと交わるその他のケーブル

4)　ケーブルの種類の混在

5)　転送速度

5)の転送速度の違いで，1)～4)のことが出てくるので，ここで1)～4)について順に考えていきましょう。

まず1)のSCSI2デバイスとSCSI1デバイスの混在です。先に示したとおり，我が家のX68030周りは，X68030以外すべてSCSI2機器です。

SCSI1からSCSI2へ変わった点を大雑把にいえば，

A)　電気的特性が厳しくなった

B)　デバイスタイプが増えた

C)　規格が標準化にともない，敷居が上げられた

D)　WIDE SCSI,FAST SCSIの追加など，新しい転送方法が追加された

ほかにも細かいことはいくつかあるのですが，関係ありそうなところはこの4つぐらいです。

A)の電気的特性が厳しくなったというのは，D)のFAST SCSI(つまりは高速同期転送)に関連することが主です。理系の人はわかると思いますが，高周波が有限の2つの直線の上を伝送されるとき，考慮しなくてはいけないことがいくつかあるからです(図6)。

そのひとつはまず電線の長さによる抵抗です。抵抗は電線の長さ，伝送される周波数，そして温度などに関係します。さらに，隣接する電線同士には周波数に応じて，コンデンサがつながっているのと同じになります。このコンデンサは，高周波になればなるほどインピーダンスが低くなります。

単純にいえば，高周波であればあるほど，絶縁されているはずの2本の電線は，「お互いの線に流れている信号同士，影響しあうようになり，結果としてノイズとなる」，「内部抵抗が増えてきて，信号が伝わりにくくなる」という現象が表れてくるということです。

SCSI1では厳密な取り決めをしてないため，高速な転送になるとたびたびエラーが起きました。こうしてSCSI2では，A)に示すとおり電気的条件が厳しくなったのです。

B)はデバイスタイプが増えただけで，この場ではあまり関係はありません。

C)の敷居の上昇は，SCSI1のときは，SCSI規格準拠と書かれていても，つながらないものがいくつかあったために行われました。

結果的にこのケース(我が家のX68030周辺)ではSCSI1とSCSI2が混在しています。SCSI2で細かに決められた電気的特性のなか，X68030だけ「SCSI2の規格から見れば」規格外ということになるのです。

イニシエータがX68030のときはSCSI2デバイスを利用していても，すべてSCSI1レベルで動いているのでエラーは出ません。しかし，CZ-6VS1はSCSI2によって動いているイニシエータです。相手がX68030ならばCZ-6VS1もSCSI互換の転送方法で転送してくれますが，相手がSCSI2-HDDだとSCSI2の同期転送を行うわけです。

「ちょっと待って。SCSI2のCZ-6VS1とSCSI2のHDDで転送されているときに，どうしてX68030が関係あるの？」

そう考えるかもしれません。そう思うのが落とし穴です。高周波であればあるほど，同じバスライン上にある，「そのときは意味のない配線」が影響を及ぼすのです。SCSI2規格で細かに決められたケーブル条件はこのようなときにも影響を及ぼさない最低条件として決められているということです。

次に2)の内蔵HDDケーブルです。このHDDケーブルは，図7に示すとおり，フラットケーブルで接続されています。ここには注意すべき点が2つあります。ひとつはターミネータに関することと，もうひとつは，スタブ長に関することです。

基本的にSCSI機器というものは，ターミネータで両端を止めます。この図を見ればわかるとおり，ターミネータを端に置くためには，本当ならばHDD側になくてはいけません。これは中間にあるターミネータを無効にし，HDD内蔵のターミネータを有効にすれば回避できます(HDDによってやり方が違いますし，ものによってはターミネータは内蔵されていないものも当然あります)。

次にスタブ長の問題です。

スタブ長というのは，SCSI装置外に出ているコネクタと，実際のなかのSCSI機器を接続するケーブル長のことです。図7では，2つのフラットケーブルの長さがスタブ長になります。

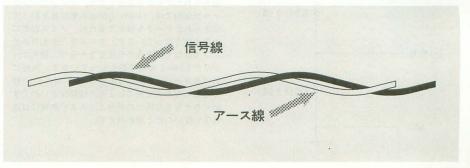
SCSI1では細かな取り決めはなかったようですが，SCSI2ではスタブ長はできる限り短く，最大10cmまでと決められています。図7でのスタブ長は約12cm＋約25cm＝37cm。思いっきりSCSI2の規格からは外れています。これではこのフラットケーブルによって作られたSCSIバスラインのせいで，おかしくなっても不思議ではありません*9。これの対処は，もはや物理的な条件なのでしょうがないですから，できるだけ信号劣化を防ぐことに頭を使います。

フラットケーブルの上で，シングルエンド型(不平衡型)SCSIバスは一般に信号ラインとGNDが交互に伝送されます。こうすることによって，隣りあった信号線が影響しあわないようにしています。影響をさらに少なくするには，図8で示したような，GNDとのツイストペア型で伝送します。

そこで，図7における真ん中のSCSIバスからHDDバスの間をツイストペアでつないでノイズ耐性を上げました。

そして3)のケーブルと交わるその他のケーブルです。

これは，たとえば電源ケーブルとSCSIケーブルが絡みあっていたり，長めのSCSIケーブルを美しく配置しようと，丸めて使ってあった場合どうなるでしょうか？　高周波であればあるほど，電流は電波に近くなりますから，ケーブルの外を突き破って近くのケーブルに影響します。そして信号劣化が起こり，結局は転送ミスとなりかねま

図8　ツイストペア線

せん。必要以上に長くてはいけないということです。最小長は30cmとSCSI2で規格づけられているので，50cmぐらいのSCSIケーブルを購入しました。

最後に4)のケーブルの種類の混在についてですが，これもSCSI2規格によって電気的特性が変わったということが原因となっています。

SCSI2規格には転送速度に関することから，ケーブルとして利用できる，最低条件が決められています。SCSIというのは50pinのケーブルでつながれていますが，要はこの50pinをなにも考えずにつないではいけないということです。

SCSI2の規格では，ケーブルはインピーダンスが90Ω以上140Ω以下であること，さらに，FAST SCSI利用時は90Ω以上132Ω以下でなくてはならないと要求されています。また，同一SCSIバス上で，異なる特性のインピーダンスを持つケーブルを混在させてはいけません。これは，異なったインピーダンスを持つケーブルを混在させると信号反射などが起き，信号の伝送特性が下がるからです。

FAST SCSI利用時はこれ以外にも，信号減衰特性，信号線間の伝搬遅延時間差，直流時の抵抗値などこと細かに決められています（スタブ長に関することは上で述べたとおりです）。

さらにシングルエンド型伝送系のSCSIの場合，ディファレンシャル型*10伝送系のSCSIと違い，高速転送時のデータ劣化が現れやすいので，ケーブル全長は6m，SCSIケーブルの長さはひとつにつき30cm以上，ツイストペア線で接続されることを推奨しています。

とはいえ，普通の人がSCSI機器を購入するときはたいていケーブルはショップ任せです。上に挙げたようなこと細かな規格をすべて網羅しているケーブルを入手できるとは限りません。ほとんどの場合は，増設したHDDなどに付属のケーブルを使うことになるでしょう。残念ながら，HDD付属のケーブルは，FAST SCSI2対応のHDDでない限り，あまり良質なものではありません。ですから，増設HDDを選ぶ際はできる限りSCSI2のHDDを選ぶことをおすすめします。

また，一般にハイインピーダンスケーブルといわれているものは，だいたいにおいて，インピーダンス100Ω前後（±10%以下）のケーブルで，ツイストペアであることが多いようです。太さも結構太いので，見分けがつくと思います。もちろん，SCSI2準拠のケーブルのことを意味しています。

SCSI1の時代，ケーブルの最小インピーダンスは90Ωよりももっと下で，ものによっては50Ωから80Ωぐらいのものもありました。同一バスライン上に異なるインピーダンスケーブルを加えてはいけないので，すでに多量にHDDをつけている人は，これらを考慮するとよいかもしれません。もっとも，ケーブルを全部変えなくてはいけませんが。

ケーブルを購入する際は，あまりに極安のジャンク品は購入しないこと，できる限りハイインピーダンスケーブルといわれているものを購入することをお勧めします。また，外付けHDDと一緒に付属品として購入する方は，できる限りSCSI 2のHDDを選ぶことをお勧めします。

＊9　とはいえ，SCSI2の規格は，高速同期転送モードでの最大値10Mバイト/秒でボローが起きないように決定されています。CZ-6VSIは2.8Mバイト/秒程度ですから，そこまで気合を入れなくてもよいのかもしれません。

＊10　ディファレンシャル型SCSI　シングルエンド型SCSIは，GND,-DB0バスとの間の電圧を計って伝えられたデータを検出します。ディファレンシャル型SCSIでは，+DB0,-DB0との電圧差を計って伝えられたデータを検出するため，ノイズなどにとても強くなっています。ターミネータもアクティブではなく，普通の分圧型で十分で，最大ケーブル長も26mと非常に長くなっています。規格におけるマージンがかなりあるので，10Mバイト/秒でももちろん大丈夫でしょう。性能はよいのですが，おそらく互換性の都合上，あまり市場には流れていないのだと思われます。

図9　シングルエンド型ターミネータ（抵抗分圧型）

アクティブターミネータ

ここまで念を入れておけば，まずSCSIバスラインは安定します。

せっかく，ここまで追求したのですし，今月は回路図も出てませんから，ついでにターミネータに関する話をしましょう。

一般にターミネータといわれるものは，図9のように，分圧によるものがほとんどです。図中，±5％とされていますが，ノイズマージンを向上させるため，できる限り1％未満であることが推奨されています。また，-信号というのは，データバスすべてのこと(-DB0,-DB1,-DB2,-DB3,-DB4,-DB5,-DB6,-DB7,-DB8,-DBP,-ATN,-BSY,-ACK,-RST,-MSG,-SEL,-C/D,-REQ,-I/O)です。信号の数だけ同じ回路が追加されます。

このようなターミネータは，単にバスラインの反射を防ぐためのもので，SCSI1を遡り，SASIの時代から，存在していたものでした。バスラインの反射を防ぐという理由のとおり，SCSIバスラインの両端に置かねばならないものです。

シングルエンド型伝送系のSCSIバスラインはもともとノイズマージンや，周波数特性が悪く，高速な伝送がしにくいといった欠点があります。これを改善するために，ターミネータもアクティブターミネータといわれる強力なタイプに改善されました。これはSCSI2から導入されています。

具体的な回路図は，図10です。

図中の低ドロップアウト型3端子レギュレータは，LT1086Cなどを使います。回路はレギュレータによってターミネータ用電源(TERMPWR)を安定させ，110Ωとケーブルインピーダンスに近い抵抗をつけることによって信号の伝送特性を上げています。

これをつけると信号ラインの安定度は格段に違います。できるのなら，X68030内蔵のターミネータは分圧型なので，これを取り外して，アクティブターミネータにすれば安定度は段違いになるでしょう。アクティブターミネータは終端だけではなく，中間に入れることもあるようです。中間に入れるタイプのアクティブターミネータ*11は，ハーフピッチのものは探してもありませんでしたが，フルピッチのものはいろい

ろなところに売っています。

XVI以前の機種をお使いの方で，SCSIバスが最近不安定だと思う方，本体SCSIコネクタに中間アクティブターミネータをつけると安定すると思われます。ハーフの中間アクティブターミネータが出るのも時間の問題だと思いますが。

＊11　SxSIというソフトによって，SASIマシンを疑似SCSIにした場合，パリティ回路付加と同時に，このアクティブターミネータを内蔵することをおすすめします。
また，自分自身でハンダづけはしない方でも，中間に差すタイプのアクティブターミネータを購入して，バスラインの間（本体SASIコネクタの直後）に挟むと信号が格段に安定します。

クロックアップの話

さて，お決まりの裏技です。

X680x0シリーズのSCSIを司るSPC(SCSI Protocol Controler)は富士通製のもので，怪しげな機能もある半面，なにしろ古い石のため，速度的にかなり不満な面があります（同期転送ができないとか）。

回路上，このICには5MHzが入っていますが，このICに入るクロックを12.5MHzまで上げてみました（実機はX68030）。

結果，最大転送レートをCONNER CP50540相手に1.5Mバイト/秒まで上げることができ，CZ-6VS1のメモリ取り込みモード160×120(6万色)で26コマも取り込むことができました。

SPCクロックはシステムクロックに影響します。システムクロックはX68030も含め，X68kシリーズは10MHzです。SPCはこれの2分周を利用するので，システムクロックを上げてあるマシンではアクセスが速くなるというのですが……。

システムクロックアップは，SCSIアクセス高速化などの効果がある半面，実際のマシン速度は遅くなることもあります。また，マシンクロックを上げることより，危険度が高いため，あまりお勧めできませんね。

まあ，ドットクロックアップとどちらが危険か？　といわれたら，ちょっと首をひねってしまいますが，あくまでもこれは挑戦としての実験でした。

あまり真似はしないほうがよいかもしれません（うちもいまは10MHzのままです）。

まとめ

今月はCZ-6VS1を絡めたX680x0周りのSCSI情報をまとめてみました。

多少難しい点もあったでしょうが，基本的には，「読み物」的な要素を表に出して書いたつもりです。いかがでしたでしょうか？

ところで，現在のうちの環境は，

ID0：QUANTUM　EMPIRE1080S　1220
ID2：CONNER　CP30540545MB3.5B0BE
ID4：TOSHIBA　OD-D300　000B
ID5：SHARP　CZ-6VS1　1.00
ID6：SONY　CD-ROM CDU-55S1.0f
ID7：SHARP　X68030

と，こんな感じです。

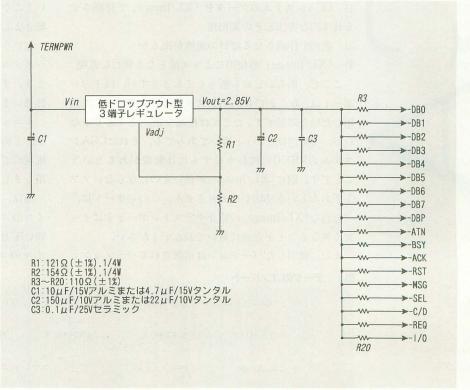
結局，小さなHDD(といっても270Mバイト，200MバイトのHDD）を友人に安く売り，さらに545MバイトのCONNER-HDDも売って，その資金でQUANTUMの1GバイトHDDを購入して内蔵HDDにしてみました。速度は割と高速で満足していますが，あとはX68030のSCSI転送レートがせめて5Mバイト/秒ぐらいになってくれればよいかなあなんて考えているところです。

ついでですが，SCSI2規格には，シングルエンド型伝送系SCSIではFAST SCSIは推奨できないことが注記されています。さらに，一般に製品を作る際は，規格の倍ぐらいまでの性能を持っておくのが普通なのですが，シングルエンド型伝送系SCSIでは規格上からして，特性マージンに余裕がないため，6mでもかなりのシビアな長さになるようです。

SCSIがらみの話をし始めると，話が尽きません。この記事を書くためになんだか，もの凄く勉強したような気がしますが，ひょっとしたらSCSIがらみのお仕事をしている人から見たら，「？」となってしまうところがあるかもしれません。致命的なミスはないとは思いますが，なにかあったらよろしくお願いします。

また，今回，ツクモ電機ニューセンター店の伊藤さんには，相当お世話になりました。文中に出ているショップというのは，ツクモニューセンター店のことです。この場を借りてお礼申し上げます。どうもありがとうございました。

参考文献
桑野雅彦，Inside X68000，ソフトバンク
桑野雅彦，Outside X68000，ソフトバンク
桑野雅彦，X68030 Inside/Out，ソフトバンク
SCSI2詳細解説，CQ出版

図10　シングルエンド型アクティブターミネータ

XL/ImageでCGA(その1)

プロジェクトチームDōGA
かまた ゆたか

せっかく発売された高機能レンダラ「XL/Image」をCGAに活かさない手はない。CGAシステムのレンダラとして使ってみよう。でも,CGAシステムのデータをどの程度利用できるのか？ 実用一点張りの緊急レポート。

はじめに

すでにご承知のように,イマジカテクノシステムより高性能レンダラ「XL/Image」が発売されました。あのパーソナルリンクスのレンダラがX680x0に移植されたということで,当チームもたいへん期待しております。これがどのようなソフトであり,どんなことができるかといった一般的な紹介は,先月号の江口響子さんのレポートや,付録ディスクに関する記事などで詳しく解説されていますので,まずはそちらをご覧ください。

この連載では,「XL/Image」を使ってCGAを作ってみる,というより,CGAシステムのレンダラ(REND)の代わりに使用してみるという実験レポートをお届けします。とはいっても,実はまだ手元に「XL/Image」が届いていません。さてどうなることやらちょっと不安です。

チェックポイントは,次の3点です。

1) CGAシステムのデータを「XL/Image」で作画させる具体的な方法とその実用性
2) 動画を作画させるだけの速度が出るか
3) 「XL/Image」の使用により可能となる新たな表現

ここで,あらかじめお断りしておきますが,以下のレポートは,あくまでもCGAシステムのレンダラとして使用した場合の話です。たとえば極端な話,「とてもじゃないが,実用性はない」と書いてあっても,それはCGAシステムのRENDの代わりにするには無理があるということです。別に「XL/Image」が使いものにならないソフトであるという意味ではありません。このレポートは,一般的な「XL/Image」の紹介やテストレポートとはまったく異なることを念頭に置いて読んでください。

また,使用したバージョンは市販されるバージョンとは異なるということや,私がまだ「XL/Image」を使いこなしていないので,正しくない表現があるかもしれませんが,ご了承ください。

データフォーマットの種類

さて,「XL/Image」が届きました。いちばん気になるのが入力するデータのフォーマットですが,CGAシステムと似ているのでしょうか。

「XL/Image」は,パーソナルリンクスのレンダラであり,そのもとになったリンクスというシステムは,もともと大阪大学工学部の大村先生のグループで開発されたシステムです。当チームも,ご存じのように大阪大学コンピュータクラブから発生しており,CGAシステムのベースであるCGA共通規格を検討したメンバーには,大村先生のグループの学生も参加しています。つまり,CGAシステムと「XL/Image」は,兄弟とはいわないまでも,いとこぐらいの関係にあるのです。知らなかったですね。驚きましたね。

ということで,「XL/Image」のフォーマットも,CGAシステムにちょっとパラメータが増えたようなものでしょう。まずはサンプルデータを見てみましょう。……全然違いますね。まったくわかりませんね。

マニュアルをよく読んで,勉強すればよいのでしょうが,どのみち,このデータを一からエディタで書くのは現実的ではありません。さっそく,DōGAコンバータを使用しましょう。

実は,このコンバータ関連のツール,初めは形状ファイルのコンバータしかなかったのですが,某方面からの強い圧力により,急きょアトリビュート,フレームファイルのコンバータも作られました。そのため,「XL/

図1 データのコンバート

CGAシステム	:形状ファイル	アトリビュート	フレームファイル	画像ファイル
	↓	↓	↓	↑
コンバータ	:SUF2PPD.X	ATR2CMD.X	FRM2CMD.X	LPC2DOGA.X
	↓ (SUF2PPD.X→コマンドファイル)	↘	↙	↑
XL/Image	:PPDファイル	コマンドファイル		LPCファイル

Image」のマニュアルの印刷に間に合わなかったということで，あまり詳しく解説されていません。ということは，この原稿がマニュアル代わりになるのかも……。

「XL/Image」で使用するデータファイルとCGAシステムのファイル，そしてコンバータは図1のような関連になっています。

まず，PPDファイルですが，これはポリゴンに関するファイル……CGAシステムの形状ファイルに相当するものです。ポリゴン以外の形状はSHPファイルで扱いますが，とりあえずこちらは関係ありません。PPDファイルと形状ファイルの違う点として留意しなければいけないことは，まず，PPDファイルでは，凹ポリゴンは扱えないということ，そして，1つのPPDファイルの中では法線ベクトルがないポリゴン(poly, uvpoly)と，法線ベクトルがついているポリゴン(shade, uvshade)を混在させることはできないという2点です。これらの問題については，ちゃんと解決策が用意されていますので，あとで詳しく解説します。

次に，コマンドファイルですが，これはだいたいフレームファイルとアトリビュートファイルを合わせたようなものです。アトリビュートファイルとフレームファイルが一緒になっているというのは，ちょっと意外なようですが，アンビエントの意味をもう一度思い出すと，確かに正しいような気もします。

また，RENDではオプションで指定するアンチエイリアシング，解像度などもこのコマンドファイル中に記述します。

データコンバートの実際(その1)

なにはともあれ，一度コンバートして，作画させてみましょう。最初から複雑なデータを使うと不幸になりそうなので，まずは非常に簡単なデータで実験してみます。

サンプルとして用意しましたのは，SAMP1.SUF, SAMP1.ATR, SAMP1.FSC です(リスト1～3)。5面から成る四角錐が，10フレームかけて一直線に動くだけというシンプルなものです。

コンバートの手順としては，

1) FFでフレームファイルを作っておく

  FF SAMP1

CGAシステムのいつもの操作です。

2) 形状ファイルのコンバート

  SUF2PPD SAMP1.SUF

と実行すると，

  P_SAMP1.PPD

  S_SAMP1.CMD

が出力されます。

3) アトリビュートのコンバート

  ATR2CMD SAMP1.ATR

と実行すると，

  A_SAMP1.CMD

が出力されます。

4) フレームファイルのコンバート

  FRM2CMD SAMP1.FRM SAMP1.SUF SAMP1.ATR

と実行すると，

  SAMP1.CMD

が出力されます。

コンバートの作業は以上です。とっても簡単じゃないですか！ 注意する点としては，4)のフレームファイルのコンバートを実行する以前に，形状ファイル，アトリビュートのコンバートを実行しておかなければならないことです。

さて，コンバートが終了すると，

  XLIMAGE

として，XL/Image を起動させます。そして，

  < SAMP1.CMD

とすると，コンバートしたコマンドファイルを読み込んで，自動的に10フレーム作画を始めます。PPDファイルなどは指定する必要はありません。

なお，コマンドラインから，

  XLIMAGE < SAMP1.CMD

のように入力すると，XL/Imageが起動して，コマンドファイルを読み込み，作画実行したあと，終了してコマンドラインに戻ります。つまり，RENDとまったく同じ感覚で使用することができます。

作画した結果が写真1で，写真2がCGAシステムによる作画です。

さて，作画された画像は，LPCファイルとなっていますので，これをCGAシステムの画像ファイルに戻さないとアニメーションできません。これは，

リスト1 SAMP1.SUF

```
obj suf samp1 {
atr teimen
prim poly  (    100      100     -100
                100     -100     -100
               -100     -100     -100
               -100      100     -100 )
atr sokumen
prim poly  (      0        0      100
                100      100     -100
                100     -100     -100 )
prim poly  (      0        0      100
                100     -100     -100
               -100     -100     -100 )
prim poly  (      0        0      100
               -100     -100     -100
               -100      100     -100 )
prim poly  (      0        0      100
               -100      100     -100
                100      100     -100 )
}
```

リスト2 SAMP1.ATR

```
atr teimen        {
                  col ( rgb ( 0.00 1,00 1.00 ) )
                  amb ( 0.3 )
                  dif ( 0.7 )
                  spc ( 0.8 0.3 0 )
        }
atr sokumen       {
                  col ( rgb ( 1.00 1.00 0.50 ) )
                  amb ( 0.2 )
                  dif ( 0.8 )
                  spc ( 0.8 0.2 0 )
        }
```

リスト3 SAMP1.FSC

```
#frame( fno, 1, 10 )
@5.3@
fram {  light pal( rgb ( 1.00 1.00 1.00 ) -3.00 -2.00 -4.00 )
        { mov ( 350  ¥div( -200 , 150 , 1 , 10 , fno )¥ 150 )
          eye deg( 60 )
        }
        { mov (  0  0  -50 ) target }
        { mov (  0  0   0 ) obj samp1 }
}
#endframe
```

LPC2DOGA SAMP1001.LPC SAMP1001.PIC

というように，1枚ずつコンバートする必要があります。結構めんどうですね(バッチファイルでもできるけど)。もっとよい手をあとで考えましょう。

データコンバートの実際(その2)

ではもうひとつ，それなりに複雑な形状をコンバートしてみましょう。以前GENIEなどで制作した宇宙戦闘機です。今回はさらにコックピット周辺にマッピングまで施しました(写真3)。形状はX2.SUF，アトリビュートはGENIE2.ATR，フレームファイルはSAMP2.FRMです。また，マッピング用の画像ファイルは，FIRE.PICです。

まず，形状ファイルのコンバートが若干異なります。

SUF2PPD X2.SUF

を実行すると，

P_X2.PPD
N_X2.PPD
S_X2.CMD

の3種類のファイルが出力されます。この「N_X2.PPD」というのは，法線ベクトルがついたポリゴンのファイルです。先にも述べたようにPPDファイルは，法線ベクトルがついていないポリゴン(poly, uvpoly)と，法線ベクトルのあるポリゴン(shade, uvshade)を混在できないため，このように自動的に2つのファイルに分類してくれるわけです。ファイル数が増えますが，フレームファイルをコンバートするときや，作画するときも特にそれを意識して操作する必要はありません。

また，PPDファイルは，凹ポリゴンを認めませんので，PPDCVX.Xによって，凹ポリゴンを複数の凸ポリゴンに変換してやる必要があります。このX2.SUFに凹ポリゴンがあるかどうか定かではないですが，ないと断言できなければこの処理を実行するべきでしょう。

PPDCVX P_X2.PPD
PPDCVX N_X2.PPD

とします。この場合，それぞれのファイルをオーバーライトすることになります。別のファイルに出力することもできますが，その場合，ファイル名が異なるため，フレームファイルのコンバートの際に不都合が生じてしまいます。

さて，アトリビュート，フレームファイルのコンバートは，まったく同じ要領です。

ATR2CMD GENIE2.ATR
FRM2CMD SAMP2.FRM X2.SUF GENIE2.ATR

しかし，これだけではいけません。マッピング用の画像ファイルもLPCファイルにコンバートする必要があります。ところが，図1のとおり，LPCファイルをCGAシステムの画像ファイルにコンバートするプログラムは

写真1 XL/Imageによる四角錐の作画例

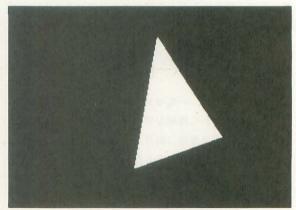



写真2 CGAシステムのRENDによる四角錐作画例

写真3 XL/Imageにより作画した宇宙戦闘機

写真4 CGAシステムにより作画した宇宙戦闘機

あっても，その逆はありません。どうしましょう。

仕方がないので，LPCSAVE.Xを使います。まず，SLIDEで FIRE.PICを表示しておいて，

LPCSAVE FIRE.LPC 0 0 256 256

とします。このプログラムは文字どおり，画面に表示されている内容をLPCファイルにセーブしてくれます。

書式は，

LPCSAVE 保存ファイル名 X始点 Y始点 X大きさ Y大きさ

です。ですから，マッピング画像のサイズが512の場合，X，Yの大きさを512にしてください。

以上でコンバート作業はできました。

XLIMAGE < SAMP2.CMD

で実行させた結果が写真3(SAMP2_XL.PIC)です(ただし，この写真は，後述のアンチエイリアスを行っています)。最後にLPC2DOGAで画像ファイルにコンバートして終わりです。

出力画像の違い

通常，レンダラが異なれば，質感が異なるのは当然として，表示される位置などにも微妙に差が出ます。これは，レンダリングアルゴリズムが異なるのに加えて，透視変換の方法が違っていたり，ハイライトや点光源の減衰を表現する式が異なるのが原因です。

ということで，皆さんも21ページのGraphic Galleryの画像をよく比べて見てください。ひと言でいうと，ほとんど同じです。誌面上の写真で差が出るのは，ハイライトのつき方だけでしょう。これは，CGAシステムがグーローシェーディングなのに対して，XL/Imageはフォンシェーディングだからです(コラム参照)。

ただ，戦闘機のコックピット部分の質感が若干異なります。XL/Imageの場合，半透明を表現するときは，レイトレース関連のパラメータをいろいろ与えないといけないのです。この問題は少し難しいので，来月号に先送りします。

次に，2つの画像データを重ねてみても，表示位置などはほぼ同じです。1カットの中にXL/Imageの画像と，CGAシステムの画像を混在させると，さすがに，ピクッと動いた感じになりますが，ここまで一致するとは少々驚きました。ただ，アンチエイリアスを実行しない場合，XL/Imageの出力画像は，CGAシステムより，少し右に表示されています。とはいっても，その差は1ドット以下で，斜辺のジャギーの出方が違うという程度です。これは，XL/Image のアンチエイリアスのアルゴリズムの問題だと思われます(コラム参照)。

あと，ディザリングアルゴリズムが異なるので質感が少し違うのですが，問題になることはほとんどないでしょう。

ちょっと困ったことは，このコンバータで作られたコマンドファイルでは，出力されるLPCファイルは，256の解像度の場合，内容に関わりなく263Kバイトになるということです。これでは，夜な夜な作画させようとしても，すぐディスクがいっぱいになってしまいます。この問題については，あとで直接CGAシステムの画像ファイルを出力する方法を検討してみましょう。

各ファイルの中身

CGAシステムの各ファイルをコンバートして作画するだけでは，XL/Imageの実力を発揮できないどころか，アンチエイリアスがかかった画像すら作画できません。そこで，XL/Imageの各ファイルの中身を見てみましょう。上記のSAMP1をコンバートしたファイルが，リスト

グーローとフォン

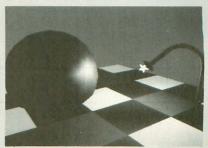
まずCGAシステムのRENDが行っているグーローシェーディングとは，ポリゴンの頂点が何色になるかだけを計算するというアルゴリズムです。ほかの部分の色は，こっちが赤であっちが青なら，その真ん中は紫だろうってな感じで，いい加減に補間しているだけです。

それに対して，XL/Imageが行っているフォンシェーディングとは，すべての場所で，ちゃんと法線などに基づいてそこが何色になるか計算して求めています。

当然ですが，フォンのほうが作画時間が何倍もかかります。しかし，作画時間をかけるだけのメリットが2つあります。1つは，ハイライトがハッキリ出る，また言い方を変えると点光源の効果が出やすいということです。典型的なケースとして，写真Aと写真Bをご覧ください(AがXL/Image，BがCGAシステム)。しかしこのメリットが，実際にCGA作品を作るうえで，そんなに大きな差となるケースはそれほど多くないでしょう。

それよりも大きなメリットといえるのは，さらに高度な表現につながるという点だと思います。すべての位置でちゃんと法線計算などをしているため，バンプマッピング，環境マッピング，映り込みなども可能なのです。

これらの表現は，グーローシェーディングを前提としているRENDには，決してできないことです。

写真A　XL/Imageによる作画例

写真B　CGAシステムのRENDによる作画例

4～7です。なお，各行の左端にある「数字：」は，解説のためにつけた行番号です。

1) N_SAMP1.PPD

まず，ポリゴンデータである「N_SAMP1.PPD」(リスト4)ですが，1～7行のヘッダ記述は，データ領域を確保するために，面数や総頂点数などを記しています。9～12行のsolidは，要するにその下のpart, vertex, faceのID(各行の左端についている数字)が1から何番までを使っているかということです。

13～16行のpartは，アトリビュートが同じポリゴンをグループ分けしています。14行目は，下のfaceのIDの1～1，つまり1だけが同じアトリビュート，15行目は2～5の4面が同じアトリビュートのグループだということです。

18～24行のvertexは頂点の座標値を，26～32行のfaceはvertexで示した頂点のどれとどれを結んでポリゴンとするかを頂点のIDで表現しています。なぜ，このように頂点とその連結情報を分けるかといえば，たとえばある頂点を4つの面が共有していたとすると(図2)，CGAシステムの場合，その頂点を4回透視変換しますが，PPDファイルのフォーマットでは，透視変換が1回ですむからです。

とまぁ，そういうフォーマットなのですが，これをエディタだけで一から作るのは大変でしょうね。できるだけ，いじりたくないファイルといえるでしょう。

2) S_SAMP1.CMD

ポリゴンデータがコンバートされるときに生成される「S_SAMP1.CMD」(リスト5)は，いったい何が書かれているのでしょう。……わかりませんね。たぶんアトリビュートとの関連について記述していると思うのですが，これを解析するにはちょっと時間がかかりそうです。とりあえず，あまり重要そうではないので，飛ばしましょう(スンマセン)。

3) A_SAMP1.CMD

アトリビュートファイルをコンバートしたのが「A_SAMP1.CMD」(リスト6)ですが，これは簡単！　一目瞭然，CGAシステムとほとんど同じです。アンビエントやディフューズの値が3つずつあるのは，RGBごとに設定ができることを意味します(CGAシステムでもできるけど)。スペキュラーに関する見慣れないパラメータとして「spc1」がありますが，この第2成分と第3成分は通常同じで，ハイライトの広がりを表しています。アトリビュートスペキュラーの第2成分と同じ役目ですが，計算式が異なるので，違う値になっているわけです。

4) SAMP1.CMD

さて，いちばん重要なのが，フレームファイルをコンバートした「SAMP1.CMD」(リスト7)です。単にフレームファイルの情報以外に，アンチエイリアスや解像度を指定するというRENDのオプションの情報もここに入っています。XL/Imageの豊富な表現を利用するときも，このファイルをもとに，いろいろ書き加えます。

1，2行目は，外部ファイルをインクルードすることを意味します。4，5行目は「obj_root」で記述される物体を，「camera」で記述される視点から見たところを作画するという意味です。つまり，コマンドファイル中には，複数の視点を記述しておいて，この「select」で選択する

アンチエイリアス

CGAシステムとXL/Imageでは，アンチエイリアスのアルゴリズムが若干異なります。

まず，CGAシステムの場合，「/A」オプションをつけないと(「/A1」でも同じ)アンチエイリアスがない状態であり，各ピクセルについて，その中心の色を求めてそのピクセルの色とします。「/A2」のときは，1つのピクセルを縦横2分割し，4倍のオーバーサンプリングをして，その平均の色をそのピクセルの色とします。「/A3」の場合，縦横3分割して，9倍のサンプリングになります(図A)。

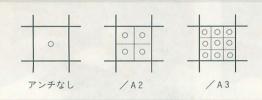
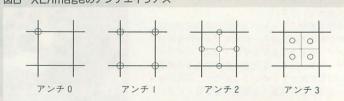
これに対してXL/Imageは，図Bのようにアンチ0の場合，ピクセルの真ん中ではなく左上の位置の色を求めています。

次に，アンチ1の場合，4隅の色を求めて平均化します。これは一見CGAシステムの「/A2」に相当しているように見えますが，それは錯覚で，アンチエイリアスではありません。なぜなら，4隅の色というのは隣のピクセルの色であり，その平均化とは，色をぼやかしているだけで，オーバーサンプリングしているわけではないからです。それが証拠に，アンチ1にしても作画スピードは変わりませんし，画質も向上しません。

アンチ2の場合は，隣のピクセルと比較し，色がほとんど同じだったら問題なしとし，もしその差が激しかった場合，図のように5カ所の位置の色を求め，平均化します。つまり5倍のオーバーサンプリングです。

さらに，アンチ3では，先ほどの5カ所の色の差が激しかった場合，さらに図の4カ所をサンプリングします。このアンチエイリアスは，CGAシステムより賢く，画質も若干よくなります。

CGAシステムの場合，「/A2」が標準的なアンチエイリアスといえるでしょう。XL/Imageでこれにいちばん近いのは，やはりアンチ2だと思います。ということで，画質テストと速度テストは，どちらもにパラメータを2の状態で実験してみました。

図A　CGAシステムのアンチエイリアス

図B　XL/Imageのアンチエイリアス

○の位置の色を計算して求めている

ことができるわけです。それがどういうメリットなのかはわかりません。

7～10行の「global」は，RENDのオプションのような役目を果たしています。7行目はアンチエイリアスの指定，8行目は作画中のメッセージの表示，9行目はエラーが発生したときの対応，10行目は背景の指定です。

13～25行が視点，注目点の設定です。作画解像度の指定や，ファイル出力名もここで与えています。

27～33行で，作画する物体，つまり4行目の「obj_root」の内容を記述しています。しかし，今回は四角錐が1つだけのはずなのに，「object」が3つもあります。なぜかというと視点(_obj_1)や注目点(_obj_2)も，「object」とみなされるからです。また，有効となる光源も記述されています。アンビエントも1つの光源として別に指定されるところが面白いですね。

34～36行では，先ほど有効にした「_obj_3」とは，「samp1」のことだと指定しています。そして「samp1」の具体的な内容は，「S_SAMP1.CMD」や「N_SAMP1.PPD」といった別ファイルに記述されます。えらく遠回しな指定ですね。

37～40行は，「_obj_2」つまり注目点の位置情報，同様に41～44行は，「_obj_1」つまり視点の位置情報を記述しています。

45～48行は，アンビエントの具体的な内容ですが，光源の種類がアンビエントということ以外指定する内容はないので，中身はありません。それに対して49～54行の平行光線は，光源の種類，色，位置，向きなどの情報が記述されています。

そして，これだけの情報を全部用意して57行目で作画しています。

REND並みの表現

CGAシステムのレンダラとしてXL/Imageを使用するなら，まずは最低限，RENDで表現できる程度のことを実現できるようにしましょう。

以下，上記の「SAMP1.CMD」(リスト7)を例にとって解説します。

リスト4 N_SAMP1.PPD

```
 1:header
 2:      solid    1
 3:      vertex   5
 4:      part     2
 5:      face     5
 6:      fnode    16
 7:end
 8:
 9:solid
10:      1          /v 1 5 /p 1 2 /f 1 5
11:end
12:
13:part
14:      1 /f 1 1
15:      2 /f 2 5
16:end
17:
18:vertex
19:1      100.000000 100.000000 -100.000000
20:2      100.000000 -100.000000 -100.000000
21:3      -100.000000 -100.000000 -100.000000
22:4      -100.000000 100.000000 -100.000000
23:5      0.000000 0.000000 100.000000
24:end
25:
26:face
27:1      1 2 3 4
28:2      5 1 2
29:3      5 2 3
30:4      5 3 4
31:5      5 4 1
32:end
```

リスト5 S_SAMP1.CMD

```
 1:global      side-free       on
 2:create      object  default samp1
 3:    side    both
 4:    object  _p1_samp1
 5:    object  _p2_samp1
 6:close
 7:
 8:create      object  default _p1_samp1
 9:    surface teimen
10:    geometry _p1_samp1
11:    smooth  off
12:close
13:create      geometry ppd     _p1_samp1
14:    file-name       p_samp1.ppd
15:    part            1 1
16:close
17:create      object  default _p2_samp1
18:    surface sokumen
19:    geometry _p2_samp1
20:    smooth  off
21:close
22:create      geometry ppd     _p2_samp1
23:    file-name       p_samp1.ppd
24:    part            2 2
25:close
```

リスト6 A_SAMP1.CMD

```
 1:create  surface  default  teimen
 2:    rgb       0.000000 1.000000 1.000000
 3:    ambient   0.300000 0.300000 0.300000
 4:    diffuse   0.700000 0.700000 0.700000
 5:    specular  0.800000 0.800000 0.800000
 6:    spcl      1 0.130714 0.130714
 7:close
 8:create  surface  default  sokumen
 9:    rgb       1.000000 1.000000 0.500000
10:    ambient   0.200000 0.200000 0.200000
11:    diffuse   0.800000 0.800000 0.800000
12:    specular  0.800000 0.800000 0.800000
13:    spcl      1 0.174738 0.174738.
14:close
```

図2 1点を4面が共有

リスト7 SAMP1.CMD

```
 1:< a_samp1.cmd
 2:< s_samp1.cmd
 3:
 4:select      object  obj_root
 5:select      camera  camera
 6:
 7:global      antialias       0
 8:global      verbose         on
 9:global      error           2
10:global      background      off
11:
12:;** FRAME 1 **************
13:create      camera  default camera
14:    pos             0 0 0
15:    tar             -1 0 0
16:    direction       +z
17:    pos-coord       _obj_1
18:    tar-coord       _obj_2
19:    perslen         1.732046
20:    proj-offset     0 0
21:    proj-size       256 256
22:    rend-offset     0 0
23:    rend-size       256 256
24:    image                   samp1001.lpc
25:close
26:
27:create      object  default obj_root
28:    object  _obj_3
29:    object  _obj_2
30:    object  _obj_1
31:    light   _amb on
32:    light   _light_1 on
33:close
34:create      object  default _obj_3
35:    object  samp1
36:close
37:create      object  default _obj_2
38:; target
39:    mov     0.000000 0.000000 -50.000000
40:close
41:create      object  default _obj_1
42:; eye
43:    mov     350.000000 -200.000000 150.000000
44:close
45:;ライト
46:create      light   default _amb
47:    type    ambient
48:close
49:create      light   default _light_1
50:    type    parallel
51:    rgb     1.000000 1.000000 1.000000
52:    pos     0 0 0
53:    tar     -3.000000 -2.000000 -4.000000
54:close
55:error-init
56:
57:view
58:
59:;** FRAME 2 **************
        以下省略
```

1) アンチエイリアス

7行目の「global antialias」で指定します。0ならアンチなし，1はあまり意味はなく，2がCGAシステムの「/a2」に相当します。特に画質を優先する場合は3にします。通常は2でよいでしょう。アンチエイリアスのアルゴリズムについては，コラムをご覧ください。

2) 高解像度

256×256ではなく，512×512の解像度の画像を作画したい場合は，21，23行目の「proj_size」，「rend_size」を以下のように変更します。

```
20：    proj-offset    0 0
21：    proj-size      512 512
22：    rend-offset    0 0
23：    rend-size      512 512
```

「-offset」というのは，画面の左上の座標です。「-size」は画面の大きさであって，画面右下の座標ではありませんので，「511」と間違わないでください。「proj」と「rend」の違いはあまり気にしないで，とにかく両方書いておけば問題ありません。

3) 出力ファイル名の指定

REND でいうところの「/O」オプションです。出力ファイル名を設定しているのは，

```
24：    image          samp1001.lpc
```

の行です。ですから，この行の最後のファイル名を書き換えれば，好きな名前で出力できます。

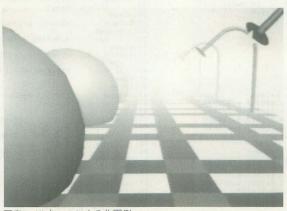
4) CGAシステムの画像出力

LPCファイルは，大きくなるうえにコンバートしないとHANIMでアニメーションできません。そこで，CGAシステムの画像ファイルを直接出力することを考えてみました。

まず，LPCファイルを出力しないように，先ほどの1行，

```
24：    image          samp1001.lpc
```

を削除します。

次に，画像ファイルの出力ですが，57行目の「view」によって，画面に作画しますので，この画面をCGAシステムのSAVEでセーブしてやればよいのです。

コマンドファイルは，「sh」というコマンドで，CGAシステムやHuman68kのコマンドが使用できます。うまくすれば，バッチファイルのような使い方もできます。

```
58：    sh save /u /l samp1001
```

という1行を加えます。すると，作画したのち，通常のRENDのように，samp1001.picを出力するようになります。「/u」「/l」というオプションをつけたのは，512の画面の左上4分の1の256の画面をセーブするからです。512の画像を作画させる場合は必要ありません。

5) 背景合成

背景合成とは，RENDの「/h」オプションのように，別に用意した画像と合成させながら作画する機能です。まず，背景合成の指定は，

```
create    volume background back_1
          grad-type     image
          file-name          ＊＊＊.lpc
close
```

というような1文(クラス)を適当な位置に設けます。次に，

```
10：global      background      off
```

を，

```
10：global      background      on
```

に書き換えます。

そして，6行目の位置に，

```
6：select      background      back_1
```

を書き加えて完成です。XL/Imageの背景合成では，作画する画像と背景画像の縦横のピクセルがまったく異なっていても可能です。しかし，そんなことをする機会はまずないでしょう。

以上で，RENDの機能はほとんど網羅したことになります。強いていえば，XL/Imageでは星空の自動発生などができませんが，これらはおまけであり，レンダラが本来持っていなければならない機能ではありませんので，これは比較の対象外でしょう。

その他，多関節構造体，空気遠近法，点光源といった機能は，なんの修正もなく，データをコンバートするだけで実行することができます(写真5，6)。

写真5　XL/Imageによる作画例

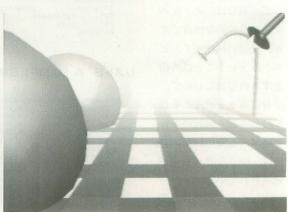



写真6　CGAシステムのRENDによる作画例

作画速度の比較

データのコンバートの次の問題は，作画速度です。ずばり，実測した値は，表1をご覧ください。

サンプルデータは，あまり特殊なケースではなく，より実用に近いデータということで，GENIEの戦闘機(マッピングあり)を1機(FIGHTER)と，霧の中の複数の球や街頭の画像(FOG)を採用しました。これらは，CGAシステムのデータをコンバートしたデータであり，球などもポリゴンで作られています。球などは2次曲面を使っていないので，XL/Imageに若干不利かもしれませんが，CGAシステムのRENDの代わりに使用するというのが今回の主題ですので，ご了承ください。

作画の条件は，RENDのオプションが「/A2」「/G」「/DE」，つまりアンチエイリアス2倍，スムースシェーディングマッピングあり，誤差拡散ディザありです。XL/Imageは，アンチエイリアスを2にしただけです。これでほぼ同じ画質になります。ただし，RENDはグーロー，XL/Imageはフォンという違いはあります(コラム参照)。

結論をいえば，XL/Imageの作画時間は，RENDの5～6倍ということになります。もっとも，画像の内容や機種，コプロの有無などの条件で値はかなり違います。

この5～6倍という速度差ですが，CGにあまり詳しくない人はXL/Imageは遅いという誤解を抱くでしょう。しかし，ちゃんとグーローとフォンの違いを理解している人なら，これはなかなかのもんだとわかるはずです。私は，先月号の響子さんのレポートの「レンダリング時間が3時間52分37秒」というのを見て，正直な話，こりゃCGAには使えないかなぁと思っていました。しかし，今回使用してみた感想としては，少なくともX68030でコプロもあれば，たいていの画像を数分で作画してくれますので，十分実用性がある速度だと思います。

表1　作画速度テスト

作画内容：FIGHTER

機種	コプロ	REND	XL/Image	比率
X68030	あり	18 秒	78 秒	4.33 倍
X68030	なし	28 秒	206 秒	7.36 倍
X68000XVI	あり	60 秒	480 秒	8.00 倍
X68000初代	なし	120 秒	790 秒	6.58 倍

作画内容：FOG

機種	コプロ	REND	XL/Image	比率
X68030	あり	44 秒	152 秒	3.45 倍
X68030	なし	62 秒	294 秒	4.74 倍
X68000XVI	あり	116 秒	720 秒	6.21 倍
X68000初代	なし	272 秒	1284 秒	4.72 倍

おわりに

以上，今回の結論は，

1)　コンバート作業は予想以上に簡単で，たいていのものがコンバートできる。ただ，アンチエイリアスなどは，若干の修正を加える必要がある

2)　作画速度は，RENDより数倍遅いが，フォンであることを考えればかなり速い

ということになります。

残念ながら，3つ目のチェックポイントであるXL/Imageならではの表現については，ページの都合で次回となります(本当は，まだ使いこなしていない)。コラムでも書きましたように，数倍の作画時間をかけてもフォンシェーディングを使うメリットは，これから真価を発揮します。ということで，次回をお楽しみに。

各読者連絡事項

●CGAコンテスト応募締め切り迫る

すでにご存じのように，第7回アマチュアCGAコンテストの締め切りが12月31日に迫ってます。応募者の皆さんは，ラストスパートがんばってください。また，昨年同様，イントロダクションのコーナーでは，数カットしかない作品や予告編，未完成作品なども紹介する予定ですので，そういった作品もお待ちしています。

<募集作品>

使用機種，ソフトは問いませんが静止画は基本的に不可です。応募形態は基本的にビデオテープとなります。

<募集部門>

一般部門，1カット部門，4カット部門

<応募方法>

まずコンテスト事務局まで「応募票」を請求し，必要事項を記入のうえ，作品とともに郵送してください。なお，応募作品の返却は行いません。

<問い合わせ先>

〒533　大阪府大阪市東淀川区淡路5－17－2
篤コーポ102号室　プロジェクトチームDōGA
「アマチュアCGAコンテスト事務局」

●賛助会員申し込み締め切り迫る

各地でバラバラにCGAの活動をしている人たちのつながりを作ろうということで，プロジェクトチームDōGAの賛助会員の名簿を作ろうというお知らせを，バージョンアップサービスの際にいたしましたが，現在のところ応募者が少なくて名簿を発行できない状態です。

たぶん皆さんお忘れになっているのだと思いますが，締め切りは12月31日ですので，マニュアルの申込書に必要事項記入のうえ，早めに送ってください。

●バージョンアップ発送終了

おかげさまで，CGAシステム2.Zのバージョンアップは，大きなトラブルもなく，すべて終了いたしました。万が一まだ届いていないという方がいらっしゃいましたら，郵便振替の控えのコピーを添えて当チームまでご連絡ください。

また，以下の方が転居先不明で戻ってきました。

・茨城県勝田市　　内田様
・神奈川県川崎市　佐藤様

さらに，不在・保管期間経過のため戻ってきた方も数件あります。当方としては，これ以上の処置はとれませんので，救済を希望されるのでしたら，ご連絡ください。

Oh!X

LIVE in '95

©COMPILE/LMS Music 1994

X68000・Z-MUSIC
Ver.2.0+PCM8用

ぷよぷよ

Tukamoto Masami
塚本 雅巳

X68000・Z-MUSIC
Ver.2.0+PCM8用

ジムノペディNO.1

Inada Hiroki・Utsumi Junichi
稲田 宏規・内海 淳一

X68000・Z-MUSIC
Ver.2.0用(SC-55対応)

PRIME

Doi Jun
土井 準

新しい年とともに，このコーナーも「LIVE in '95」となりました。今年はどんな作品が登場するでしょうか。曲づくりの気分も一新して，常連さんも初挑戦者もがんばってくださいね。入力専門の人も，感想などお待ちしています。今年もよろしく。

やったなぁー，ぎょえーん

今月の1曲目は，塚本氏作の「ぷよぷよ」のテーマです。オリジナルから大幅にグレードアップしたサンバ調のリズムが，聴くものをウキウキさせてくれます。リストを見るとサクサクっと作ってしまった感がありますが，ナカナカの完成度といえるでしょう。

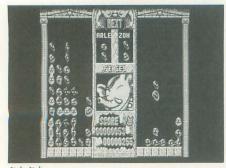
演奏にはPCM8.XとZ-MUSICver.2.0に付属のPCMデータが必要です。

そして，この曲の演奏には2つのZPDを用意しなければなりません。ひとつは今回リスト掲載されているZPD作成用CNFファイルです。そしてもうひとつはZ-MUSICver2.0のディスク1に収録されているSTD_SET.CNFです。

この両方を，

A>ZPCNV filename

としてZPDを作成してください。

Z-MUSICver.2.0にはこのほかにもCNFファイルがありますので，興味のある方は参考にしてください。

なお，今回この「ぷよぷよ」のテーマで使用されているSTD_SET.CNF(ZPD)は最も基本的なリズムキットですので，ハードディスクに保存しておいて損はないでしょう。特に，試験的に曲を作る場合などは新たにZPDを作り起こさなくても，これさえあればだいたいこと足りるでしょう。

基本的なものは揃っていますので，曲の制作の最終段階で，リズムキットを元にオリジナルのZPDを作る……という手法が効率的でいいと思いますよ。

現代リズムに邂逅するサティ

2曲目はちょっと変わっています。きわどい和音と独創的なリズムが特徴のサティのピアノ曲，これのグルーヴィーアレンジバージョンをお届けしちゃいます。クラシック音楽をその時代時代に流行している形態へアレンジする試みは世界中で行われてきました。日本ではクライズラー&カンパニー(以下K&K)などが有名ですね。

でも，こういうアレンジ曲は原曲のあらたな魅力を引き出しているケースとそうでないケースがあります。K&Kの「亡き王女のためのパヴァーヌ」は素晴らしいアレンジですが，同じK&Kの「ユーモレスク」はちょっとやりすぎって感じです。

今回掲載する稲田/内海両氏の共作による「ジムノペディNO.1」はやりすぎない程度の，実に聴き心地のよいアレンジで好感がもてます。あの，葉から朝つゆが滴り落ちるような静かな曲が，なんともコクのあるグルーヴに変身しているさまは，一聴の価値ありです。もうちょっとベースラインを他パートと絡みあうように刻んでもよかったと思いますが。

演奏にはPCM8.XとZPD作成のために，Z-MUSICver.2.0ムックに収録されているPCMデータが必要になります。

ぷよぷよ

サティ：ピアノ作品集

R・E・S・O・R・T

さわやかフュージョンでリフレッシュ

今月最後はT-SQUARE(曲発表当時はTHE SQUARE)の「PRIME」です。作者の土井氏はこの曲に大変な思い入れがあるそうですが，なるほど大変よくできています。パート別に聴くと確かにメロディやバックパートの一部，リズムの音色(これはしかたないですが)の感じがCDとは違うのですが，全パートまとめて聴くとまったく異和感はありません。

実をいうと，この曲はほかにも数人から投稿されていました。そんななかでも土井氏の作品は，いちばん各楽器の演奏がバランスよくミキシングされていて，とてもいい仕上がりをみせていました。そこで，文句なしに採用となったのです。

エフェクタの設定も完璧，DTMでは難しいとされるギターソロもカッコよく決まっています(ちょっとハーモニクスがくどい気もしますが)。

演奏にはSC-55系GS音源が必要です。こちらで試してみたところSC-55/SC-55mkII両方で正常な演奏を確認しました。内蔵音源は使用していないのでミキサーなどは不要です。

リストは途中まで打ち込めばそこまでの全パートが演奏される形式です。土井氏もいっていますが，入力と試聴を交互に繰り返せば入力ミスを発見しやすく，入力する側にとっては親切な記述方式といえます。

データの記述には，人それぞれのスタイルがありますが，長めの曲のリストには，こういった書式が向いているかもしれませんね。　(Z.N)

リスト1　ぷよぷよ

```
  1: .comment Theme of PUYOPUYO by Masami 94/10/17
  2:
  3: / 制作 : Masami
  4: / 協力 : 八千代
  5: / 資料提供 & 応援 : なこりん
  6: / 参考 : ALL ABOUT ぷよぷよ
  7: / 初版 93/12/26  第2版 94/10/17
  8:
  9: .adpcm_block_data=STD_SET.ZPD
 10: .adpcm_block_data=PUYO.ZPD
 11:
 12: (i)
 13:
 14:                          / 68SND.ZMS から持ってくればよい
 15: (v8,0
 16: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 17:           58, 15,  2,  0,210,  0,  0,  0,  0,  3,  0
 18: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 19:           31, 13,  1,  4, 15, 41,  2, 15,  3,  0,  0
 20:           31, 20,  5, 15, 14, 57,  1, 13,  7,  2,  0
 21:           20, 10,  1,  7,  8, 35,  1,  3,  7,  0,  0
 22:           23,  5,  1,  7, 15,  0,  0,  1,  3,  0,  1)
 23:
 24:                          / 68SND.ZMS から持ってくればよい
 25: (v10,0
 26: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 27:            3, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0
 28: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 29:           31, 14,  1, 10, 10, 42,  0,  6,  6,  0,  0
 30:           31,  5,  0, 10,  6, 26,  0,  0,  4,  0,  0
 31:           31,  2,  4,  6,  1, 32,  0,  0,  4,  0,  0
 32:           28,  1,  6,  8,  1,  0,  0,  1,  3,  0,  1)
 33:
 34:                          / 打ち込もう!
 35: (v71,0
 36: /         AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
 37:           61, 15,  0,  1,  2,  0,  0,  0,  0,  3,  0
 38: /         AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
 39:           18,  7,  5,  5,  2, 28,  0,  2,  0,  0,  0
 40:           31,  2,  5,  7,  2,  3,  0,  1,  0,  0,  0
 41:           31,  2,  5,  7,  2,  2,  0,  2,  0,  0,  0
 42:           20, 10,  5,  7, 10,  2,  0,  4,  0,  0,  0)
 43:
 44:
 45: (m1,1024)(aFM1,1)         (m2,1024)(aFM2,2)
 46: (m3,1024)(aFM3,3)         (m4,1024)(aFM4,4)
 47: (m5,1024)(aFM5,5)         (m6,1024)(aFM6,6)
 48: (m7,1024)(aFM7,7)         (m8,1024)(aFM8,8)
 49:
 50: (m 9,1024)(aADPCM1, 9)  (m10,1024)(aADPCM2,10)
 51: (m11,1024)(aADPCM3,11)  (m12,1024)(aADPCM4,12)
 52: (m13,1024)(aADPCM5,13)  (m14,1024)(aADPCM6,14)
 53: (m15,1024)(aADPCM7,15)  (m16,1024)(aADPCM8,16)
 54:
 55:
 56: / Temp = 120
 57: (o120)
 58:
 59: (t1,2,3) @8 v11 @s5 @m10 @a5 @h0
 60: (t1) o5 p3
 61: (t2) o4 p1
 62: (t3) o3 p2
 63: (t6) @10 v13 q7 p3 o2
 64:
 65: (t4,5) @71 @s5 @m10 @a5 @h12
 66: (t4) v13
 67: (t5) v11 r16 @k5
 68:
 69:
 70: (t1, 2, 3, 4, 5, 6, 7, 8) [do] 18
 71: (t9,10,11,12,13,14,15,16) [do]
 72:
 73:
 74: (t1)     |:13c+c+c+c+c+c+c+c+dddddddd:|
 75:          c+c+c+c+c+c+c+c+ f+g4g4g4g
 76:
 77:          rf+f+f+f+f+f+f+ g+4f+e4e4.
 78:          rf+f+f+f+f+f+4 >b4<eg+4f+4.
 79:          rf+f+f+f+f+f+f+ g+4f+e4dc+d
 80:          e1 d4.e4.c+4
 81:          e1 d4.e4.c+4
 82:
 83:
 84: (t2)     |:216a:|                         / 27小節間
 85:          <df+4f+4f+4f+>
 86:
 87:          raaaaaaa b4ag+4g+4.
 88:          raaaaaa4 g+4g+g+4f+4.
 89:          raaaaaaa b4ag+4g+f+g+
 90:          a1 g+4.a4.f+4
 91:          a1 g+4.a4.f+4
 92:
 93:
 94: (t3,6)   |:14 aabb<ccc+c+  ddf+f+ddee  :|
 95:
 96:          |:3  ddeef+f+g+g+ eef+f+g+g+bb:|
 97:          f+f+f+f+f+f+f+f+ ddddeeee
 98:          f+f+f+f+f+f+f+f+ dddddddd
 99:
100:
101: / Melody
102: (t4,5) r1r1r1r1
103:        |: o5
104:          |:r8c+4>ab4ab<ddc+>bab4.<   r8c+4>ab4ab f+f+ed&d2:|
105:          e<e4>e<e4>e<e> e<e>e<e4e>e4 e<e4>e<e4>e<e
106:          |1>e<eeeeeee
107:        :||2>e<e4e4e4e
108:
109:           rddddddd e4dc+4>b4.
110:          <rdddddd4>g+4b<e4d4.
111:           rddddddd e4dc+4>bab
112:          <c+1 >b4.<c+4.>a4
113:          <c+1 >b4.<c+4.>a4
114:
115:
116: / Snare & Bass Drm.
117: (t9) l16o2
118:          c8c8d8cdcdccd8cd c8c8d8cdcdccd8c8
119:          c8c8d8cdcdccd8cd c8c8d8cd cd8c<cc>ba
120:      |:
121:          |:2 c8c8d8cdcdccd8cd  c8c8d8cdcd8cddcc
122:              c8c8d8cdcdccd8cd  c8c8d8cdcd8c<cc>ba :|
123:          |: ccccdccdcdccdccd cdccdccdccdcdccc :|
124:      :|
125:      |:3 ccc8 d8c8 c8cc dcc8 ccdc c8cd cdcd c8dd :|
126:      |:  c8dc cdcd ccdc dcd8 ccdd cdcc ddcd ccdc :|
127:
128:
129: / Crash Cym & etc.
130: (t10) r1r2.o1 v11 d+8d+8 r1r2.d+16d+16d+8
131:      |:
132:          v9o3c+1 r2.o5v7b4&b1 r2.o3v7c+4
133:          v9  c+1 r2.o5v7b4&b1 r2.o3v8c+4&
134:          o2l8|:v9d+v10d+v11d+rv9d+v10d+v11d+r
135:                o6v6cv7cv8cv10cv8cv9cv11cr:|
136:      :|
137:          v9o3c+1r2.o5v7b4o3v9c+1r2.o5v7b4
138:          o3v9c+1r1 |:o3c+1o6b4.<c+4.>a4:|
139:
140: / Hi-hat
141: (t11) o2l16 |:38
142:          v10f+v8f+rv10f+ v8f+rv10f+v8f+
143:          rv10f+v8f+r v10f+v8f+rv9f+:|
144:
145: / Tamb.
146: (t12) o3l16 |:38
147:          v5f+v3f+v6f+8v9f+v4f+v6f+v5f+
148:          v6f+v3f+v4f+v6f+v9f+8v7f+v8f+:|
149:
150: / Cow-bell
151: (t13) o3l4 |:38 v6g+v7g+v6g+v7g+:|
152:
153: / Agogo
154: (t14) o4v7l16 |:38
```

```
155:         v7g+8v6g+8v10g8v6g+v7g+
156:         v8g+v10gv5g+v6g+v9g8v7g+v6g+:|
157: / Bongo
158: (t15) o4l16
159:         |:38 v5c+8c+8v7c8v5c+v5c8.v9c+8v4cv5c+v6c+8:|
160:
161:
162: / Timb. & Hand-Clap
163: (t16) l16
164:     o2v9r4d+4r4d+4 r4d+4r4d+8d+8
165:        r4d+4r4d+4 r4d+4o4r{v6fv7fv8f}v7fffff8
166:     |:
167:        o2v9r4d+4r4d+4 r4d+4ro4v9f8v8f+v9ffv8f+8
168:        o2v9r4d+4r4d+4 r4d+4ro4v9f8v8f+r{v6fv7fv8f}v9fv8f
169:        o2v9r4d+4r4d+4 r4d+4r4d+8d+8
170:           r4d+4r4d+4 r4d+4
171:        o4r{v7fv8fv9f}v8ff
172:          r{v7f+v8f+v9f+}v8f+f+
173:        |:o4
174:          |:v7f+8f+8{v4fv5fv6f}8v5ff:|
175:          |:{v6fv7fv8f}8v7ff:|
176:         v6f+v7f+v9f+8v8{fff}8v7f+8:|
177:     :|
178:
179:     |:3 rv9f+8v4f+                  v10fv8fv7f+8
180:        r{v5fv7fv9f}v7fv8f          v6fv5fv6f+8
181:        v7f+v10f+8v8f               v9f+v8f+v7f8r{v5fv7fv9f}
182:        v8fv5f                      v7fv9fv8f8:|
183:
184:       |:2 r{v4fv5fv6f}v4ff          r{v5fv6fv7f}v5ff
185:           r{v7fv8fv9f}v7ff          r{v9fv10fv11f}v9ff
186:           rv10f+8v8f                v9f+v8f+v7f8
187:           r{v5fv7fv9f}v10fv8f       v10fv8fv9f8:|
188:
189:
190: (t1, 2, 3, 4, 5, 6, 7, 8) [loop]
191: (t9,10,11,12,13,14,15,16) [loop]
192:
193: (p)
```

このプログラムは，株式会社コンパイルと株式会社エル・エム・エス・ミュージックの許諾により掲載したものです。個人で楽しむほかは，無断で第三者に譲渡したり転用してはいけません。

リスト2　ぷよぷよの音色コンフィグファイル

```
/ Theme of PUYOPUYO

.o6c = WHSTL.PCM
.o6a = OH4A.PCM,v65
.o6b = OH4B.PCM,v65
.o7c+= OH4C#.PCM,v65,p12
```

リスト3　ぷよぷよのカウンタ表示

```
 1:00000000 00001C80   2:00000000 00001C80   3:00000000 00001C80   4:00000000 00001C80
 5:0000000C 00001C80   6:00000000 00001C80   7:00000000 00000000   8:00000000 00000000
 9:00000000 00001C80  10:00000000 00001C80  11:00000000 00001C80  12:00000000 00001C80
13:00000000 00001C80  14:00000000 00001C80  15:00000000 00001C80  16:00000000 00001C80
```

リスト4　ジムノペディNO.1

```
  1: .comment Gymnopedie No.1  Composed by E.Satie  arr. by H.INAT
A  1993/ 4/13,14
  2:
  3: / 1993/4/13,14 arranged by H.INATA for ZMUSIC.X
  4: / 1994/1/26,6/28 rearranged by J.UTSUMI for ZMUSIC.X + PCM8.X
  5: / INTERNAL
  6:
  7: /----- ADPCM DATA SET -----------------------------------------
  8:
  9: .adpcm_block_data=gymno1.ZPD
 10:
 11: /----- TRACK SETUP ---------------------------------------------
 12:
 13: (i)
 14: (o96)
 15:
 16: (m01,2000)(aFm1,01)
 17: (m02,2000)(aFm2,02)
 18: (m03,2000)(aFm3,03)
 19: (m04,2000)(aFm4,04)
 20: (m05,2000)(aFm5,05)
 21: (m06,2000)(aFm6,06)
 22: (m07,2000)(aFm7,07)
 23: (m08,2000)(aFm8,08)
 24: (m09,2000)(aAdpcm,09)
 25: (m10,2000)(aAdpcm,10)
 26: (m11,2000)(aAdpcm,11)
 27:
 28: /----- OPM DATA SET --------------------------------------------
 29:
 30: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME      E.Bass1
 31: (@30,      31, 11,  8,  7,  5, 28,  0,  2,  7,  0,  0
 32:     31,  9,  4,  7,  5, 34,  0,  4,  3,  0,  0
 33:     31, 14,  2,  7,  2, 26,  0,  1,  3,  0,  0
 34:     26,  7,  3, 11,  5,  0,  0,  2,  7,  0,  0
 35: /       AL  FB  OM
 36:      1,  7, 15)
 37:
 38: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME      E.Bass2
 39: (@31,      31, 10,  4,  9,  4, 30,  0,  2,  7,  0,  0
 40:     31,  8,  6,  7,  3, 26,  0,  1,  7,  0,  0
 41:     31, 12,  3, 14,  2, 36,  0,  2,  3,  0,  0
 42:     26,  7,  3, 11,  5,  0,  0,  2;  3,  0,  0
 43: /       AL  FB  OM
 44:      3,  7, 15)
 45:
 46: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME      Synth1
 47: (@32,      31,  0,  0,  0,  0, 24,  0,  4,  7,  0,  0
 48:     18,  7,  3,  7,  3,  5,  0,  4,  7,  0,  0
 49:     31,  0,  0,  0,  0, 17,  0,  2,  3,  0,  0
 50:     26, 11,  2,  7,  3,  0,  0,  2,  3,  0,  0
 51: /  AL  FB  OM
 52:          4,  7, 15)
 53:
 54: /  AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME      Synth2
 55: (@33,      31, 11,  4,  6,  3, 28,  0,  4,  7,  0,  0
 56:     18,  9,  5,  8,  3,  3,  0,  4,  7,  0,  0
 57:     31, 13,  3,  6,  3, 22,  0,  2,  3,  0,  0
 58:     31, 13,  2,  8,  3,  0,  0,  2,  3,  0,  0
 59: /  AL  FB  OM
 60:          4,  7, 15)
 61:
 62: /  AR 1DR 1DR  RR 1DL  TL  RD MUL DT1 DT2 AME      CHORD
 63: (@34,      12,  0,  2,  0,  0, 44,  0,  1,  3,  0,  0
 64:     17,  0,  0,  7,  0, 10,  0,  2,  7,  0,  0
 65:     14,  0,  2,  0,  0, 48,  0,  8,  7,  0,  0
 66:     19,  0,  3,  7,  0,  6,  0,  1,  3,  0,  0
 67: /  AL  FB  OM
 68:     4,  7, 15)
 69:
 70: /----- MML DATA SET --------------------------------------------
 71:
 72: / OPM
 73:
 74: (t1)        q812@k0 @a6@s,4 o2 [do]
 75:    v14 |:8@30d4<@31d>>@30a4<@31a:|
 76:    v15 @30<c+.f+.>b.b.a.e.a.a.a.a.a.a.a.a.a.
 77:    b.<c+.>f+.b.b.b.e.a. [loop]
 78:
 79: (t2)        q814v13@k2 @m8@s6@h22 @32p3o5 r*6 [do]
 80: (t3)        q814v11@k-4 @m16@s6@h22 @32p1o5 r*42 [do]
 81: (t4)        q814v14@k-1 @m8@s4@h22 @33p3o5 [do]
 82: (t5)        q814v12@k-7 @m16@s4@h22 @33p2o5 r*36 [do]
 83:
 84: (t2,3,4,5)
 85:    r1r1r1
 86:    rc+edc+>g+f+g+ae2. c+2.< |:3r2.:|
 87:    rc+edc+>g+f+g+ae2. g+2.<c+2.
 88:    >>b2.< r2.r2. ef+gbaf+agf+a2. r2a
 89:    b<cde>gabaf+a2.
 90:    r2 a<d2.c+2. >f+ef+g+abg+abe2. r2.r2. [loop]
 91:
 92: (t6)        q812.v12@k4 @m8@s5@h36 @34o4 [do]
 93:    |:8f+e:|
 94:    >g+<f+ >f+f+ ag bb gg <e>b
 95:    bg g<b e>a g+g+ r4f+4f+4 ga< [loop]
 96:
 97: (t7)        q812.v12@k1 @m8@s5@h36 @34o4 [do]
 98:    |:8ag+:|
 99:    c+a d>a< c+>b <dd> b<c+ ge
100:    d>b< c+f+ g+f+ >b<c+ r4>a4a4 b<c+ [loop]
101:
102: (t8)        q812.v12@k-2 @m8@s5@h36 @34o4 [do]
103:    |:8<c+c+>:|
104:    ea dd ee ba ee ge
105:    de ef+ g+f+ ee r4>a4b4< de [loop]
106:
107: / ADPCM
108:
109: (t9)        l16v9 o3 [do]
110:    |:8rrrrd8rd8_12d8~rrrrrd8rrd8r_d~:|
111:    |:4r8rrd8rdrdr8:|
```

▶定期購読にしてから，Oh!Xが届く頃になるといつもきまって雨が降る。おかげで手にするのはふにゃふにゃのOh!X……。ああ！　雨男(女)は誰だ!?……って，俺か？　（それにしても付録ディスクは心配だ）　味野　真一(28)岡山県

```
112:    |:6rrrrd8rd8_d8~r|rrrrd8rrd8r_d~:|
113:    |:12f8c8:|
114:    |:4rrrre8re8r|_e8~:|_e~e [loop]
115:
116: (t10)   l16v9 o3 [do]
117:    |:32~8c8_20c8~12c8:|
118:    |:4~8c8_20c~8c8_c~c8_c8~12c8:|
119:    |:22~8c8_20c8~12c8:|
120:    |:12g8_24g8~:|
121:    c8.cc8ccc8cc|:3c8ccc8ccc8cc:| [loop]
122:
123: (t11)   l16v9 o2 [do]
124:    |:16cd8ccd8cd8cd:|
125:    f4d8ccd8d8 e8.cd8ccd8d8 e8.cd8d8d8d8 e8.cd8ccd8d8
126:    e8.ccd8cd8cd|:10cd8ccd8cd8cd:|
127:    f4|:11e4:|
128:    ~8e8._ccd8cd8cd|:3cd8ccd8cd8cd:| [loop]
129:
130: (p)
```

リスト5 ジムノペディNO.1の音色コンフィグファイル

```
 /   Set Rhythm
/   [ gymno1.CNF ]

1 = elcs_.pcm, v40

.o2c = CH1.pcm, v140
.o2d = OHH808.pcm, v200
.o2e = TRIANGLE.pcm
.o2f = crash11.pcm, v140

.o3c = U110BD1.pcm, v70
.o3d = TR909sd.pcm, v90
.o3e = sd808.pcm, v77, m1
.o3f = WDSD1.pcm, v60, m1
.o3g = SIDE.pcm, v90

.erase 1
```

リスト6 ジムノペディNO.1のカウンタ表示

```
 1:00000000 000015F0   2:00000006 000015F0   3:0000002A 000015F0  4:00000000 000015F0
 5:00000024 000015F0   6:00000000 000015F0   7:00000000 000015F0  8:00000000 000015F0
 9:00000000 000015F0  10:00000000 000015F0  11:00000000 000015F0
```

リスト7 PRIME

日本音楽著作権協会(出)許諾第942410-401号

```
  1: / ZMS>55>prime_sc.Zms
  2: /=============================================================
  3: /                          P R I M E
  4: /
  5: /               安藤まさひろ／T H E  S Q U A R E
  6: /               アルバム "R・E・S・O・R・T" より
  7: /
  8: / 参考：『T-S Q U A R E   B E S T』(リットーミュージック刊)
  9: /        Oh!X 1993.7 掲載 田辺正則さん制作「MIDNIGHT CIRCLE」
 10: /         その他各プログラム
 11: /=============================================================
 12: / version 0.99z     1993-06-29(TUE)
 13: / version 1.00      1993-09-12(SUN)
 14: / version 1.01      1994-01-15(SAT)
 15: / LIVE掲載版        1994-10-11(TUE)
 16:
 17: / このデータは掲載用に編集し直した物で、演奏自体はNETにUPされて
 18: / いる物と同じですのでお持ちの方は入力しなくても良いです。
 19: / 使用音源はCM-500(mode C)ですが、SC-88でも聴けます(音が柔らか
 20: / くて原曲に近い感じ♪)。
 21:
 22: /                                          N obbie
 23: /=============================================================
 24:
 25: .comment  =PRIME= Masahiro Ando/THE SQUARE "R・E・S・O・R・T" by N obbie
 26:
 27: (i)
 28: (o152)
 29:
 30: (d0)
 31:
 32: (m 1,3000)(aMIDI10, 1)     / Bass Drum
 33: (m 2,3000)(aMIDI10, 2)     / Snare & Tom
 34: (m 3,3000)(aMIDI10, 3)     / Cymbal
 35: (m 4,3000)(aMIDI 1, 4)     / Bass
 36: (m 5,3000)(aMIDI 2, 5)     / Guitar 1
 37: (m 6,3000)(aMIDI 3, 6)     / Guitar 2
 38: (m 7,3000)(aMIDI 4, 7)     / Synthe 1
 39: (m 8,3000)(aMIDI 5, 8)     / Synthe 2 & Guitar (delay)
 40: (m 9,3000)(aMIDI 6, 9)     / Piano
 41: (m10,3000)(aMIDI 7,10)     / Melody
 42: (m11,3000)(aMIDI 8,11)     / Melody (delay)
 43:
 44: /=============================================================
 45:
 46: .sc55_init
 47: /.Roland_exclusive $10,$42 ={ $00,$00,$7f,$00 }    / SC-88
 48:
 49: .SC55_V_Reserve ={ 1,2,2,4,3,4,2,2,0, 4, 0,0,0,0,0,0 }
 50:
 51: .SC55_Reverb ={ 3,3,0,100,70,100,0 }
 52:
 53: .SC55_Chorus ={ 3,2,90,8,90,5,20,0 }
 54:
 55: /.Roland_exclusive $10,$42 ={ $40,$00,$04, 70 }    / volume 0-127
 56:
 57: /=============================================================
 58:
 59: / トラック10と11は基本的に同じデータなのでコピーした方が楽
 60: / だと思います。あと他にも似たような所があるのでどんどん手抜き
 61: / して下さい
 62:
 63: (t1)         @1   L4  q8 o1 @v112 @u115 @p64 @r1 @k0  r2
 64: (t2)              L4  q8 o2       @u107 @p64 @r1      r2
 65: (t3)              L4  q8 o2       @u100 @p64 @r1      r2
 66: (t4)         @34  L8  q8 o1 @v115 @u115 @p64     @k0  r2
 67: (t5)      i8 @31  L8  q8 o2 @v105 @u115 @p54     @k-1 r2
 68: (t6)      i8 @28  L8  q8 o2 @v110 @u100 @p74     @k1  r2
 69: (t7)         @100 L8  q8 o4 @v100 @u90  @p84     @k-2 r2
 70: (t8)         @101 L16 q8 o5 @v100 @u87  @p64     @k-3 r2 r*1
 71: (t9)         @3   L8  q8 o4 @v100 @u85  @p44     @k2  r*95
 72: (t10)        @66  L8 @q2 o4 @v123 @u118 @p64     @k0  r2
 73:           @m18 @h96
 74: (t11)        @66  L8 @q2 o4 @v123 @u85  @p64     @k3  r2
 75:           @m18 @h96 r8. r*2
 76:
 77: (t1)        @i$41,$10,$42 @g12 @e70,10
 78: (t4)        @i$41,$10,$42 @g12
 79: (t5)        @i$41,$10,$42 @g12
 80: (t6)        @i$41,$10,$42 @g12
 81: (t7)        @i$41,$10,$42 @g12
 82: (t8)        @i$41,$10,$42 @g12
 83: (t9)        @i$41,$10,$42 @g12
 84: (t10)       @i$41,$10,$42 @g12
 85: (t11)       @i$41,$10,$42 @g12
 86:
 87: / I n t r o ------------------------------------------------
 88:
 89: (t1)        |:|:12b:| b.b.|b:|r
 90: (t2)        |:|:6rd:| rd|rd:|r8d64d16..d64d8...
 91: (t3)        |:|:12a+:| <c+.c+.>|a+:|r
 92: (t4)        |:|:|:7b:|e | |:7f+:|a:| f+4.g4ab<c+>:|
 93: (t5)        |:|:b<f+<c+>b<c+>bf+>e | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|
 94: (t6)        |:|:b<f+<c+>b<c+>bf+>e | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|
 95: (t7)        |:'a1<df+' 'a1<c+e' 'f+1b<d' 'f+4.a<c+''g8^2b<d':|
 96: (t8)        |:8r1:|
 97: (t9)        |:'f+1b<df+''e1a<c+e''d1a<c+d' 'c+4.f+a<c+''d8^2gb<d':|
 98: (t10)       |:8r1:|
 99: (t11)       |:8r1:|
100:
101: (t1)        b.b.b8b.r2. brbr8b8&
102: (t2)        r1 r1 L16rd<c>ad<c>af<c>affL4d
103: (t3)        <c+.c+.c+8c+.r2. r2..c+8&
104: (t4)        <d4.e4.f+g1^1>g&
105: (t5)        @m12@h24d4.e4.'c+f+'@b0,32,240'd1^1g'>@b0@m0q5@u110 g&
106: (t6)        d4.e4.'c+f+''d1^1g'r
107: (t7)        @u+10<d4.e4.>'a<c+f+''b8^1^1<dg'@u-10
108: (t8)        |:3r1:|
109: (t9)        @u+10'd4.<d''e4.<e''f+<f+''g1^1<g'@u-10
110: (t10)       |:3r1:|
111: (t11)       |:3r1:|
112:
113: / A --------------------------------------------------------
114:
115: (t1)        [$]
116: (t2)        [$]
117: (t3)        [$]
118: (t4)        [$]
119: (t5)        [$]
120: (t6)        r1 [$]
121: (t7)        [$]
122: (t8)        r1 [$]
123: (t9)        [$]
124: (t10)       [$]
125: (t11)       [$]
126:
127: (t1)        |:32b:|
128: (t2)        |:16rd:|
129: (t3)        c+>|:31a+:|
130: (t4)        |:15g:|f+ |:15f+:|g |:15g:|f+ |:12f+:|ef+ga
131: (t5)        |:15g:|f+ |:15f+:|g |:15g:|f+ |:12f+:|ef+ga
132: (t6)        |:7r1:|
133: (t7)        @91 @q3 @u115
134:     <|:r1 r1 r4'f+a<d'r'ea<c+'r4'f+4a<d'
135:        'f+a<d''ea<c+'r'f+a<d'|'ea<c+'r'dgb':|r4.>
136: (t8)        |:7r1:|
137: (t9)        @u95
138:     |:'d1gb' 'd1gb' 'c+1ef+a' | 'a1<c+ef+':| 'a8^1<c+ef+'
139: (t10)       |:r4b@u-25a@u+25b@u-25g@u+25a<e4 dc+>b<c+dc+>a4. r2.
140:      | r1:| r2<d@u-10c+@u+10>b@u-5a@u+5
141: (t11)       |:r4b@u-25a@u+25b@u-25g@u+25a<e4 dc+>b<c+dc+>a4. r2.
142:      | r1:| r2<d@u-10c+@u+10>b@u-5a@u+5
143:
144: / B --------------------------------------------------------
145:
146: (t1)        |:28b:| b.b.r
147: (t2)        |:14rd:| rd8rd8d16<c16>a16a16
148: (t3)        <c+|:22b:|r8c+8^ bbb c+r8c+r8r>
149: (t4)        bbb<b>bbbb <|:c+c+c+<c+>c+c+<c+>c+:| ddd<d>dd<d>d
150:     d+d+d+<d+>d+d+<d+>d+ eee<e>ee<e>f4 ff<f>ff<f>f f+4.e1
```

▶PINBALLは思っていたよりも，すごくよくできていますね。すごい！

山本　勝(22)神奈川県

```
151: (t5)         |:8b:| <|:16c+:| |:8d:|
152:     |:8d+:| |:7e:|f4 |:7f:| f+4.e1
153: (t6)         @u105 rbg<d>b2 r<c+>a<ec+2 rec+ae2 rd>a<f+d2
154:     rd+>b<f+d+2 eg4bg2 rg+e+<c+>g+2 r1
155: (t7)         @49 q8 @u100
156:     >'g1<d' 'a1<e' 'a1<e' 'a1<df+'
157:     'b1<f+' 'b2..<eg'<'c+8^1g+' 'c+1^4.f+a+'
158: (t8)         |:7r1:| a+g+f+c+>a+f+<a+g+f+c+>a+g+f+c+f+g+
159: (t9)         >'g1<d' 'a2<e''a1<c+e' 'a2<c+e' |:'a2<df+':|
160:     'b1<f+' 'b2<eg''b4.<eg'<'c+8^1g+' 'c+1^4.f+a+'
161: (t10)        g2^g<d>b <c+1 c+2c+>a<er d2.r4 d+2d+>b<f+d+
162:     ef+4g4a4@u+2g+^2^@u+2e+g+b @u+3a+1^4.
163: (t11)        g2^g<d>b <c+1 c+2c+>a<er d2.r4 d+2d+>b<f+d+
164:     ef+4g4a4@u+2g+^2^@u+2e+g+b @u+3a+1^4.
165:
166: (t1)         b.bb. b.b.r r2.r8b8
167: (t2)         <c16>a16f16f16dr2 rdrd8d8 r.d8d8d8d8r8
168: (t3)         rr8a+8r8<c+. c+r8c+.r r2..c+8
169: (t4)         d+rd4. c+4.>g4.f+f+ r1
170: (t5)         d+rd4. c+4.>g4.f+f+ r1
171: (t6)         r1 i0 @30 o4 @u110 @p64 r1 r4.f+af+b<c+
172: (t7)         'd+f+b'r'c+4.f+a'
173:     'e4.g+b''f+b<d''f+4b<d'@u+10|:'b<c+f+':|@u-10 r1
174: (t8)         a+g+a+<c+>f+<a+bf+def+af+d>af+
175:     eg+b<eg+b>df+b<df+b@u+10|:'b8<c+f+':|@u-10 r1
176: (t9)         '>b<f+b'r'c+4.f+a'
177:     '>b4.<eg+''d4.f+b'@u+10|:'f+b<c+f+':|@u-10 r1
178: (t10)        @u-2br@u-2f+4.
179:     @u-23e4.@u-10d4.@u-10c+@u-10c+ @u+50r4.f+af+b<c+
180: (t11)        @u-2br@u-2f+4.
181:     @u-23e4.@u-10d4.@u-10c+@u-10c+ @u+50r4.f+af+b<c+
182:
183: / C -----------------------------------------------------------
184:
185: (t1)         |:60b:|
186: (t2)         |:30rd:|
187: (t3)         c+>|:59a+:|
188: (t4)         |:|:7b:|f+ |:7f+:|g |:7g:|a |:8a:|
189:       |:7b:|f+ |:7f+:|g |:8g:| ||:8a:|:|
190: (t5)         |:|:7b:|f+4|:6f+:|g4|:6g:|a4|:7a:|
191:       |:7b:|f+4|:6f+:|g4|:7g:| ||:8a:|:|
192: (t6)         |:d4ec+4>a4<c+4 d4>b4f+4a4 gr4.f+ra4 grf+af+b<c+
193:       d4ec+4>a4<c+4 d4>b4f+4a4 gf+g^2 | r4.f+af+b<c+:|
194: (t7)         @91 @u110
195:     |:'b2..<df+' 'a1<c+e' 'g1b<d' 'a8^1<c+e'
196:       'b2..<df+' 'a1<c+e' 'g1b<d'|'a8^1<c+e':|
197: (t8)         |:15r1:|
198: (t9)         >|:'a4.<df+''a<df+''a2<df+' 'a4.<c+e''a<c+e''a2<c+e'
199:        'g1b<d' 'a1<c+e'
200:        'a4.<df+''a<df+''a2<df+' 'a4.<c+e''a<c+e''a2<c+e'
201:        'g1b<d'|'a1<c+e':|
202: (t10)        |:d4ec+4>a4<c+4 d4>b4f+4a4 gr4.f+ra4 grf+af+b<c+
203:       d4ec+4>a4<c+4 d4>b4f+4a4 gf+g^2 | r4.f+af+b<c+:|
204: (t11)        |:d4ec+4>a4<c+4 d4>b4f+4a4 gr4.f+ra4 grf+af+b<c+
205:       d4ec+4>a4<c+4 d4>b4f+4a4 gf+g^2 | r4.f+af+b<c+:|
206:
207: (t1)         [tocoda]
208: (t2)         [tocoda]
209: (t3)         [tocoda]
210: (t4)         [tocoda]
211: (t5)         [tocoda]
212: (t6)         [tocoda]
213: (t7)         [tocoda]
214: (t8)         [tocoda]
215: (t9)         [tocoda]
216: (t10)        [tocoda]
217: (t11)        [tocoda]
218:
219: / D -----------------------------------------------------------
220:
221: (t1)         bbb.b8^ rrr rrrr8b8 |:3brrr rrrr8b8:|
222: (t2)         rdr8<c8>a |:8r1:|
223: (t3)         a+a+a+r8<c+8^> |:7g+:| |:3<c+>|:7g+:|:|
224: (t4)         |:7a:|b^1^1 a1^1 b1^1 a1^1
225: (t5)         r2..q8b^1^1 a1^1 b1^1 a1^1
226: (t6)         r2>f+edc+^1^1 |:6r1:|
227: (t7)         'a8^2..<c+e'>'a8^1^1<c+e' 'g1^1b<d' 'a1^1<c+e''g1^1b<d'
228: (t8)         r1 L8 o5 @u60 @p14
229:     |:<e>abf+aef+c+ e>b<c+eab<c+e
230:       d>abf+gef+d e>b<d>a|<def+a:|>b<dgb
231: (t9)         'a1<c+e' @101 o5 @u63 @p114
232:     |:aef+c+e>b<c+>a bf+ab<c+eab
233:       aef+de>b<c+>a bf+aef+a<d|e:|f+
234: (t10)        r2>f+ed @u-15c+^1^2c+6d6e6 d12&e12&d12^4^2^4 r4d6e6f+6
235:     e1^2 e6d6c+6 d12&e12&d12^4^2^1
236: (t11)        r2>f+ed @u-15c+^1^2c+6d6e6 d12&e12&d12^4^2^4 r4d6e6f+6
237:     e1^2 e6d6c+6 d12&e12&d12^4^2^1
238:
239: (t1)         brrr8b8 brrr r1 br2.
240: (t2)         r1 r1 L16|:d<c>af:||:<c>aff:| r1 L4
241: (t3)         |:<c+>g+g+g+:| r1 <c+2..>a+8
242: (t4)         g1 f+1 <L16gbagf+agf+edc+edc+>ba L8g2.ra
243: (t5)         g1 f+1 <L16gbagf+agf+edc+edc+>ba
244:     @m20@h24 L8g2. @m0 @u115 ra
245: (t6)         r1 r1 r1 i8 @28 o2 @u100 @p74 r2.ra
246: (t7)         <'d1gb' 'c+1f+a' r1 r1
247: (t8)         <d>abf+aef+d c+ef+a<c+ef+a
248:     @u+20L16gbagf+agf+edc+edc+>ba g2..L8r
249: (t9)         aef+de>b<d>b ab<c+e>f+a<c+e>
250:     @u+20L16gbagf+agf+edc+edc+>ba g2.. @3 L8 o3 @u90 @p44 r
251: (t10)        r1 r1 r1 r1
252: (t11)        r1 r1 r1 r1
253:
254: / E -----------------------------------------------------------
255:
256: (t1)         |:|:3b8b8r2r8b8:| | bb.b.:| brbb
257: (t2)         |:|:6rd:| | rdrd:| rdd8L16dddddd L4
258: (t3)         |:|:12a+:| | <c+r8c+.>a+:| a+a+a+r
259: (t4)         |:bbr2re f+f+r2ra bbr2re f+4.g4ab<c+>:|
260: (t5)         |:|:b<f+<c+>b<c+>bf+>e
261:          | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|> q5 @u110
262: (t6)         |:|:b<f+<c+>b<c+>bf+>e
263:          | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|
264: (t7)         |:7r1:| r2 @49 o4 @u75 @p84 r2
265: (t8)         |:8r1:|
266: (t9)         |:'b<f+''b<f+'r2r'eb' 'f+<c+''f+<c+'r2r'a<e'
267:         'b<f+''b<f+'r2. 'f+4.<c+''g8^2<d':|
268: (t10)        |:8r1:|
269: (t11)        |:8r1:|
270:
271: / F (Guitar solo) ---------------------------------------------
272:
273: (t1)         |:32b:|
274: (t2)         |:16rd:|
275: (t3)         <|:32b:|
276: (t4)         |:16g:| |:16f+:| |:16e:| |:16f+:|
277: (t5)         |:16g:| |:16f+:| |:16e:| |:16f+:|
278: (t7)         'd1^1f+a' 'c+1^1ea' '>b1^1<dg' 'c+1^1ea'
279: (t9)         @u90|:|:'b4.<dg''b<dg''b2<dg':|
280:          | <|:'c+4.ea''c+ea''c+2ea':|>:|
281:     <|:'c+4.ea''c+4ea''c+ea''c+4ea':|>
282: (t10)        |:8r1:|
283: (t11)        |:8r1:|
284: (t6)         i0 @30 L16 q8 o3 @v117 @u117 @p60 @k-1
285:     @u+5@b0,-8192,108g1^2@b0
286:     @u-19(e12,g)&r12@u+7(f+12,a)&r12<@u+7e6
287:     @m20@h24c+2^8@m0c+dedc+>b <dc+>bagbagf+edf+edc+>a
288:     @m30b4.@m0<c+8d8e8f+8g8
289:     b8<d8>a64&b16..<df+ga(g,f+)&regf+e
290:     dc+>b<dc+>bagbagf+edc+e @m30f+2..@m0@u-7d8@u+7
291: (t8)         i0 @30 L16 q8 o3 @v107 @u87 @p68 @k4 r16.
292:     @u+5@b0,-8192,108g1^2@b0
293:     @u-19(e12,g)&r12@u+7(f+12,a)&r12<@u+7e6
294:     @m20@h24c+2^8@m0c+dedc+>b <dc+>bagbagf+edf+edc+>a
295:     @m30b4.@m0<c+8d8e8f+8g8
296:     b8<d8>a64&b16..<df+ga(g,f+)&regf+e
297:     dc+>b<dc+>bagbagf+edc+e @m30f+2..@m0@u-7d8@u+7
298:
299: (t1)         |:30b:|b8b.
300: (t2)         |:15rd:|rd64d8...
301: (t3)         |:31b:|r
302: (t4)         |:16g:| |:16f+:| |:16e:| |:16f+:|
303: (t5)         |:16g:| |:16f+:| |:16e:| |:16f+:|
304: (t7)         <'d1^1f+a' 'c+1^1ea'> 'b1^1<dg' <'c+1^1ea'>
305: (t9)         'b4.<df+a''b<df+a''b2<df+a'
306:     'b4.<df+a''b4<df+a''b4.<df+a'
307:     <'c+4.ea'|:3'c+ea':|'c+4ea'
308:     'c+2^ea''c+ea''c+4ea'> 'b2<dg'r'b4.<b'
309:     <'c+2<c+''d2<d' 'e4.a<c+e''e8^2a<c+e'
310:     @L2o8c+>bagf+edc+>b agf+edc+>bag
311:     f+edc+>bagf+e dc+>bagf+edc+ r8 r2
312: (t10)        |:8r1:|
313: (t11)        |:8r1:|
314: (t6)         e&f+af+&gbaf+af+&gbaf+af+& gbaf+af+&gbaf+af+&gbaf+
315:     af+&gbg&a<c+>a&b<d>b&<c+ec+&df+
316:     af+&ga<c+e8d8>b<dc+>bab8
317:     @m70b2@m0<L12(e,f+)&rf+6a6 f+6(e,f+)&rf+6a6f+6(e,f+)&r
318:     f+6a6f+6@b0,1365,0a8&@b1365,0a8@b0f+ec+
319:     ec+>b<c+>ba@m70b2@m0> L16
320: (t8)         e&f+af+&gbaf+af+&gbaf+af+& gbaf+af+&gbaf+af+&gbaf+
321:     af+&gbg&a<c+>a&b<d>b&<c+ec+&df+
322:     af+&ga<c+e8d8>b<dc+>bab8
323:     @m70b2@m0<L12(e,f+)&rf+6a6 f+6(e,f+)&rf+6a6f+6(e,f+)&r
324:     f+6a6f+6@b0,1365,0a8&@b1365,0a8@b0f+ec+
325:     ec+>b<c+>ba@m70b2@m0> L16
326:
327: / G -----------------------------------------------------------
328:
329: (t1)         bb2r bbbb b.b.r8.b16 rr16b8.r8.b16r
330: (t2)         rd8rd8L16dd<c>a <cc>afdd<a>f<cc>af<cc>af
331:     r4d8r4d<cc>ffr d<c>affrd<c>affrd64d8...
332: (t3)         c+r8c+r8r r1 c+r8c+r8r r1
333: (t4)         |:b4.b^2^1:|
334: (t5)         @m18@h24 q8|:'b4.<f+''b8^2^1<f+':|
335: (t7)         @100 @u100 'b4.<dg''b8^2^1<dg' 'a4.<c+f+''a8^2^1<c+f+'
336: (t9)         L8 o4 @u100 'g4.b<dg''g8^2^1b<dg'
337:     'f+4.a<c+ef+''f+8^2^1a<c+ef+'
338: (t10)        |:4r1:|
339: (t11)        |:4r1:|
340: (t6)         @u-20b4&a&f+&b8&d&r&d&c+&>b&a&
341:     @u+10 @b0,-8192,72@m25b8^1@b0
342:     @m30@u+10b2^16 @32@u+20 |:3@b0,2048,0b.&@b2048,0b8:|
343:     @b0(b8^32<d)&r2^8 @u-20 @m0
344: (t8)         @u-20b4&a&f+&b8&d&r&d&c+&>b&a&
345:     @u+10 @b0,-8192,72@m25b8^1@b0
346:     @m30@u+10b2^16 @32@u+20 |:3@b0,2048,0b.&@b2048,0b8:|
347:     @b0(b8^32<d)&r2^8 @u-20 @m0
348:
349: (t1)         b.b.r b8b8b8b8b8b8r b.b.r r2.r8b8&
350: (t2)         L4rdr8d8d8d8 {rdddd<c>}{rddrdd}{rddrdd}d rdrr
351:     r8|:6u+3d64d16..:|u-18r8
352: (t3)         c+r8c+r8r r1 c+r8c+.r c+8|:6r8:|c+8&
353: (t4)         b4.b^2^1 b4.b^2^2.. g&
354: (t5)         'b4.<f+''b8^2^1<f+'
355:     'b4.<f+''b8^2^2..<f+' @m0 q5 @u110 g&
356: (t7)         'b4.<dg''b8^2^1<dg'
357:     'b4.<df+''b8^2^1<df+' @91 @q3 @u115
358: (t9)         'g4.b<dg''g8^2^1b<dg' 'f+4.a<c+df+''f+8^2^2..a<c+df+'
359: (t10)        |:4r1:| @u118
360: (t11)        |:4r1:| @u85
361: (t6)         @30o5(a8,b)&r8&a+8&@m80b8^2^2@m0 a6b6<c+6
362:     <>(c+8,d)&r8c+8@m80(c+16,d)&r16^2^2@m0
363:     L32o3g&f+&e&d&c+&>b&a&g& f+&e&d&c+&>b&a&g&f+&
364:     e&d&c+&>br8
365:     r4
366:     i8 @28 L8 q8 o2 @v110 @u100 @p74 @k1
367:     r2
368: (t8)         @30o5(a8,b)&r8&a+8&@m80b8^2^2@m0 a6b6<c+6
369:     <>(c+8,d)&r8c+8@m80(c+16,d)&r16^2^2@m0
370:     L32o3g&f+&e&d&c+&>b&a&g& f+&e&d&c+&>b&a&g&f+&
371:     e&d&c+&>br8
372:     r4
373:     @101 L16 q8 o5 @v100 @u87 @p64 @k3
374:     r4.^32
375:
376: (t1)         [d.s.] [coda]
377: (t2)         [d.s.] [coda]
378: (t3)         [d.s.] [coda]
379: (t4)         [d.s.] [coda]
380: (t5)         [d.s.] [coda]
381: (t6)         [d.s.] [coda]
382: (t7)         [d.s.] [coda]
383: (t8)         [d.s.] [coda]
384: (t9)         [d.s.] [coda]
385: (t10)        [d.s.] [coda]
386: (t11)        [d.s.] [coda]
387:
388: / Coda --------------------------------------------------------
```

▶苦痛だろうと思っていたが，やはり苦痛だった，普免取得。教官に「ちゃうなあ」といわれるたびに「うう，バイクなら……」と思ってしまって，反省してないので，一向にうまくなりません。マイッタ。　　平　勝久(21)大阪府

```
389:
390: (t1)        o1bbbb
391: (t2)        o2rdr8|:6d16:|
392: (t3)        o2a+a+a+r<
393: (t4)        |:8a:|
394: (t5)        |:8a:| q8 @u115
395: (t6)        o4r2f+ed i8 @28 q8 o2 @u100 @p74 @k1 r
396: (t7)        'a8^1<c+e' o4 @u95 @p84 @k-2
397: (t8)        r1
398: (t9)        'a1<c+e' o4 @u90 @p44 @k2
399: (t10)       o5r2f+ed
400: (t11)       o5r2f+ed
401:
402: / H (Synthe solo & Fade out) --------------------------------
403:
404: (t1)        |:32b:|
405: (t2)        |:16rd:|
406: (t3)        c+|:31b:|
407: (t4)        |:|:|:7b:|e | |:7f+:|a:| f+4.g4ab<c+>:|
408: (t5)        |:|:b<f+<c+>b<c+>bf+>e
409:         | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|>
410: (t6)        |:|:b<f+<c+>b<c+>bf+>e
411:         | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|>
412: (t7)        |:'b1<df+' 'a1<c+e' 'f+1b<d' 'f+4.a<c+''g8^2b<d':|
413: (t8)        |:8r1:|
414: (t9)        |:'f+1b<df+' 'e1a<c+e'
415:        'd1a<c+d' 'c+4.f+a<c+''d8^2gb<d':|
416: (t10)       @v110 @u120 @m22@h288b&|:16~2b8&:||:16_6b16&:| |:4r1:|
417:     @86 L12 o5 @v85 @u100 @p60 @k2 @m40 @h24 r1
418: (t11)       @v110 @u85  @m22@h288b&|:16~2b8&:||:16_6b16&:| |:4r1:|
419:     @86 L12 o5 @v85 @u100 @p68 @k-2 @m40 @h24 r2.^16
420:
421: (t1)        |:64b:|
422: (t2)        |:32rd:|
423: (t3)        |:c+|:31b:|:|
424: (t4)        |:4|:7b:|e |:8f+:| |:15g:|a:|
425: (t5)        |:4|:b<f+<c+>b<c+>bf+>e
426:          | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|>
427: (t6)        |:4|:b<f+<c+>b<c+>bf+>e
428:          | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|>
429: (t7)        |:16r1:|
430: (t8)        |:16r1:|
431: (t9)        |:4'f+1b<df+''e1a<c+e''d1gb<d' 'c+4.f+a<c+''d8^2gb<d':|
432: (t10)       @b0,1365,0e2^12&@b1365,0e12&@b0e6d6 c+6d6e3c+6>a6^4
433:     @b0,1365b8&@b1365,0b8&@b0b2^2 a6b6<c+6
434:     @b0,683c+8&@b683,0c+8&@b0c+8>a4e4.f+8e8
435:     @b0,1365a8&@b1365,0a8&@b0a2
436:     e4d8c+8>b8<c+4>b8< c+8d8e8f+8 a8e8f+8a8
437:     bab<c+>b<c+dc+dede f+ef+af+ab2
438:     <c+4d8>b8^2^6 ae16d16c+16e16dc+dc+>b<c+ L16
439:     >f+ef+af+a<c+ed12c+12d12e12f+12a12 b4.af+a8f+8ec+>b<c+
440:     >b8<a8b4<a4b8a4 g4f+4e4d8
441: (t11)       @b0,683,0c+2^12&@b683,0c+12&@b0c+6>b6 a6b6<c+3>a6e6^4
442:     @b0,1365g8&@b1365,0g8&@b0g2^2 e6g6a6
443:     @b0,1365e8&@b1365,0e8&@b0e8c+4>b4.a8g8
444:     <@b0,683c+8&@b683,0c+8&@b0c+2
445:     >g4f+8e8c+8>b4<c+8 f+8g8a8b8 <c+8>a8b8<c+8
446:     dc+daeabab<c+>b<c+ dc+dedef+2
447:     f+4f+8g8^2^6 f+c+16>b16a16<c+16>babaga L16
448:     dc+df+df+a<c+>b12a12b12<c+12d12f+12 e4.rre8r8rrrr
449:     r8f+8g4<f+4g8f+4 e4d4c+4>b8
450:
451: (t1)        ¥8
452: (t2)        ¥8
453: (t3)        ¥8
454: (t4)        ¥8
455: (t5)        ¥8
456: (t6)        ¥8
457: (t7)        ¥8
458: (t8)        ¥8
459: (t9)        ¥8
460: (t10)       ¥8
461: (t11)       ¥8
462:
463: (t1)        |:64b:|
464: (t2)        |:32rd:|
465: (t3)        |:c+|:31b:|:|
466: (t4)        |:4|:7b:|e |:8f+:| |:15g:|a:|
467: (t5)        |:4|:b<f+<c+>b<c+>bf+>e
468:          | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|>
469: (t6)        |:4|:b<f+<c+>b<c+>bf+>e
470:          | f+<c+f+e4dc+>a:| f+4.g4ab<c+:|>
471: (t7)        |:16r1:|
472: (t8)        |:16r1:|
473: (t9)        |:4'f+1b<df+''e1a<c+e''d1gb<d' 'c+4.f+a<c+''d8^2gb<d':|
474: (t10)       |:64c+>baf+<:|
475: (t11)       |:64f+rrr:|
476:
477: /===========================================================
478:
479: (p)
```

リスト8　PRIMEのカウンタ表示

```
1:00007C20 00000000  2:00007C20 00000000  3:00007C20 00000000  4:00007C20 00000000
5:00007C20 00000000  6:00007C20 00000000  7:00007C20 00000000  8:00007C21 00000000
9:00007C1F 00000000 10:00007C20 00000000 11:00007C22 00000000
```

投稿に関する留意点

Oh!X編集部にはたくさんの投稿作品が届いています。そのなかから掲載作品を選ぶわけですが，なかには，非常にハイレベルながら涙をのんでボツ，というものもあります。まず，ページの都合上，64字詰めで700行を超える大作掲載が難しいのが現状です。そのほか，著作権の関係で掲載許可が下りにくい曲もあります。特に，パソコン通信のネット上ですでに発表している作品はその旨をお書き添えください。

また，誌面では，リストは64～68字詰め程度で掲載されていますので，できれば1行は64字以内にしてください。コメント行についても同様です。採用の可否とは直接関係ありませんが，多くの人に入力して聴いてもらうためにも，読みやすいリストの記述をお願いします。

▼ (善)の「勝負はこれからだ」 ▲

POWER PCのお話

POWER PCのインストラクションの本を読んだ。RISCチップでありながら，ずいぶんと命令数が多い。680x0系とは違った発想の機能を実現する命令もあったりしてこれならアセンブラによる開発もできるんじゃないの，なんて思ってしまった。ただRISCだけあってアドレッシングやオペランドの種類によって同一機能でも命令語が違うというのが面白かった。

同じRISCで日立のSH2のインストラクションも軽く読んだんだけど，これもCISC並みの命令の豊富さ。SH2といったらSEGE SATURNのMPU。こちらはもともと68000の互換チップを作っている日立だけあって命令構造が実に680x0ライク。これならば68系アセンブラでゲーム作ってた連中もすんなり移行できちゃいそう。もしかしたら680x0系のソースをちょっと直すだけでいいんじゃないのってくらい似てる。

まーいずれにせよ，(出るかどうかはわからないけど)次期Xもアセンブラでハードが叩ける，骨の随までしゃぶれるパソコンであってほしいです。

(RISCとは……装備命令語数を少なくし，チップの回路を簡略化して命令の処理時間の短縮化を実現したものだ。命令が少ない代わりにレジスタの数が多い。演算は極力レジスタ間で行うようにし，メモリの読み書きは極力少なくするようにプログラミングする必要がある。このためプログラムはほとんどCのような高級言語によって書かれる)

容量インフレの話

その昔，8ビットパソコン時代にゃ2D(容量320Kバイト)のフロッピーディスクがあれば，ちょっとしたプログラミングやデータ収拾にゃ事足りたもんだが，いまじゃ容量128MバイトのMOがあっという間に埋まっちまう。秋葉原にいけば120Mバイト，240Mバイトのハードディスクがカゴに山積みで格安セールだ。昔のゲームは「ALLマシン語」とか書いてあっても，たかが10数キロのプログラムだったんだが，いまやゲームソフトを買えば「ハードディスクの空き容量が17Mバイト以上必要です」とか書いてあったりする。10年前の常識の約1000倍がいまの常識ってわけ。

先日，大手メーカーが1.3Gバイトの3.5インチMOドライブの開発に成功との新聞発表があった。製品となるのはまだ先だろう。しかし容量だけじゃなくてアクセス速度も従来の2～3倍とか。うげげ，CD-ROM約2枚分の容量ではないか。CD-ROM違法コピーとかそのうち問題になってくるのかしらん。歴史は繰り返すのか。うーむ。

ゲーム誌の話

某パソコン通信ネットで，某ゲーム誌の「FM TOWNS版のスーパーストリートファイターIIが移植版のなかでいちばんデキがいい」という記述に対する反論が殺到していた。ふーむ，どれだその雑誌は。見てみるとその雑誌はゲーム機とパソコンをターゲットとした総合ゲーム誌なのだが，表紙に記載されているパソコンの種類にはFM TOWNSはあってもX68000はなかった。

しかし，「ふーむ，じゃ仕方ないじゃないか」と考えてしまう自分に気づいたことが，いちばん悲しかった。　(西山善司)

▶寒いね，ヤダね，この季節。でも，秋から冬にかけての夜空は神秘的でいいやね。みんなも，つい見惚れ過ぎてボクのように風邪なんてひいちゃ駄目だよ。
太田　志輝(18)北海道

TEX入門講座 ～てふてふらてふ～

レイアウトを考えよう

Taki Yasushi 瀧　康史

今回をもって，とりあえず終了となります
今月は文書スタイルや記号など細かな部分について考えていきましょう
次の一歩はあなた自身で試行錯誤しながら踏み出してください

なにを思ったのか，いきなりレーザープリンタを買ってしまった。中古だけどね。EPSONの「LP-1500」というちょっと昔のレーザープリンタだが，前の人が相当大事に使っていたのか，品物としてはほとんど新品同様に近いお買い得品。

おかげで狭い我が家には「IO-735X」「MJ-700V2C」「LP-1500」と，２台のインクジェットと１台のレーザープリンタという，３つともでっかいプリンタがある始末。とはいえ，どのプリンタも一長一短で，どれも捨てられないなあという，困った状態。あえて処分するなら「IO-735X」だけど，黒い下着のセクシーな金髪ねーちゃんや，むきむき兄貴のポスターをでっかく作りたいからなぁ。

「LP-1500」は300dpi(スムージング機能により疑似600dpi)だけれど，やっぱり印刷結果は「MJ-700V2C」よりも綺麗。出力そのものもレーザーのほうが綺麗みたい。当然，TeXもばっちり綺麗に印刷できるし，なによりも速い速い。LP系はいずれもESC/Page(EPSON standard cord for Page printer)というEPSONの規格なので，TeXだってなにもしなくても，LP系の環境ファイルをそのまま使えるからね。しかも，『X680x0TeX』(以下TeX本)のLIPS3 DVI(ESCPDVI)をコンパイルして使えば，もうモウレツ綺麗。当然SX-WINDOWだってばっちり。SXのほうはデフォルトの環境ではだめだけど，「X-DTP」に入っているESC/Page用のデバイスドライバ(とはいえ，SYSTEM.LBに追加するリソースなんだけど)を追加してしまえば，「シャーペン」でも「Easydraw」でも印刷ができるのがいいね。「X-DTP」で同人誌だってOKだ。X680x0との相性は良好かな？

余談が過ぎた。私が執筆するTeX入門は今回が最後なので，レイアウトなどの細かいことについて話していく。

文書スタイルについて

まず最初に文書スタイルに関する話をする。連載が始まって以来，使ったdocument styleは，jarticleだけだ。LaTeXではこれ以外にも，以下のスタイルがある。

・英文用

article	学会論文誌の形式
report	テクニカルレポート
book	１冊の本の形式
letter	手紙の形式

・日本語用

jarticle	日本語学会論文誌の形式
jreport	日本語テクニカルレポート
jbook	日本語の１冊の本の形式

といった具合。

ただし，LaTeXはもともと欧米で作られたものなので，それをそのまま日本語に移植したj～の３つの形式には，多少，日本人のセンスからは逸脱しているところもあるような気がする。

たとえば，私がいままで使っていたjarticleは，大学のレポートにしても問題ない形式である。

reportという名前にひかれて，一度はjreportを使ってみたが，これはちょっと「大層」すぎる。まあ，レポート書きは現役ともいいにくいので，レポートを書くセンスも忘れてしまったのだが……。

今回はjbookスタイルを使ってみよう。

まずリスト１の17～26行目を見てわかるとおり，部，章，節，小節，小々節の区切りは，以下の命令を使って表記する。

¥part {タイトル}	部
¥chapter {タイトル}	章
¥section {タイトル}	節
¥subsection {タイトル}	小節
¥subsubsection {タイトル}	小々節

このなかで，¥partはjbookでしか使えないし，¥chapterはjbookとjreportでしか使えない。すなわち，いつも使っているjarticleでは¥section以下しか使えないということだ。

¥partは，印字例１を見てもわかるように，かなり大胆な作りになっている。まるでタイトルみたいで，次に白紙がきて(出力されなかったが，ページ数としてカウントされていた)，¥part以降の行に書かれた内容が次のページ(印字例２)に出力される。恐ろしいことに，次も白紙(ページ番号の入った紙が出力された)で，そのあとやっと¥chapterがやってくるというレイアウトだ(印字例３)。この¥chapterも通常文字に比べるとかなり大きい気がする。jreportもそうだが，jbookにしても相当厚い本じゃないと，しっくりこないようなレイアウトになる。このあたりは実際に出力して自分で確かめてみてほしい。

また，章や節を制御するコマンドは，jbookスタイルが一番多く使えるのだが，概要(abstract)を記述するabstract環境は，jreportとjarticleでしか使えない。このabstract環境は，

```
¥maketitle
¥begin {abstract}
　なんらかの概要
¥end {abstract}
```

と，記述する。わざわざ¥maketitleが書いてあるのは，この環境は必ず¥maketitle命令の次に，なくてはならないからだ。この

印字例1（70%に縮小，印刷にはLP-1500を使用）

Part I

これは部 (part) です

印字例2（70%に縮小）

part の次にくる文章はこの位置にきます。

印字例3（70%に縮小）

Chapter 1

これは章 (chapter) です

chapter の次にくる文章はこの位置にきます。

1.1 これは節 (section) です

section の次にくる文章はこの位置にきます。

1.1.1 これは小節 (subsection) です

subsection の次にくる文章はこの位置にきます。

これは小々節 (subsubsection) です

subsubsection の次にくる文章はこの位置にきます。

印字例4（70%に縮小）

LaTeX における
レイアウトに関する話から
スタイルや環境に関する話、
そして、細かな設定まで......

著：瀧　康史[1]

平成 6 年 11 月 10 日 (木)

[1] PCVAN ID:X1CLUB　　handle:紅呪

文章を実際に出力すると，jarticleの場合もjreportの場合も目次の前にくる。articleではタイトルの下，jreportでは1ページが割かれているので，前書きといった感じだろうか？　今回は使用していないので実際に試してほしい。

いい忘れていたが，jbookなどのスタイルを使うと，とても大きなフォントを利用するため，TeX本のディスクから普通にインストールした状態では，フォントが足りなくなる。同書のVol.1, 220ページ以降をよく読み, ¥METAFONTのmakefont.xを利用して，フォントを作ってほしい。

リストでは前後するが，タイトルを多少凝った作りにしてある（3～8行目，印字例4）。ただ，¥titleのなかで改行を入れたり，¥authorのなかで¥thanks{}を入れたりしているだけだが，参考になれば幸いである。また，リスト1の12～15行目は目次に関するものだが，1回目に説明したのでここでは省略させていただく。忘れてしまった方は読み直してほしい。

リスト1

```
 1: ¥documentstyle[a4j]{jbook}
 2:
 3: ¥title{¥LaTeX における¥¥
 4:         レイアウトに関する話から ¥¥
 5:         スタイルや環境に関する話、¥¥
 6:         そして、細かな設定まで$¥cdots ¥cdots$}
 7: ¥author{著:瀧 康史¥thanks{PCVAN ID:X1CLUB ¥hspace{1cm} handle:紅呪}}
 8: ¥date{平成6年11月10日(木)}
 9:
10: ¥begin{document}
11:
12: ¥maketitle
13: ¥tableofcontents
14: %¥listoftables
15: ¥vfill¥eject
16:
17: ¥part{これは部(part)です}
18: partの次にくる文章はこの位置にきます。
19:   ¥chapter{これは章(chapter)です}
20: chapterの次にくる文章はこの位置にきます。
21:    ¥section{これは節(section)です}
22: sectionの次にくる文章はこの位置にきます。
23:      ¥subsection{これは小節(subsection)です}
24: subsectionの次にくる文章はこの位置にきます。
25:       ¥subsubsection{これは小々節(subsubsection)です}
26: subsubsectionの次にくる文章はこの位置にきます。
27:
28: ¥newpage
```

記号に関するお話

普通は最初に書くべきことなのだが，今回までのTeX入門講座は「触れてみる」ことに重点を置いたので，このあたりの細かいことはあと回しにしてきた。

しかしながら，このまま説明せずに終わってしまうのもまずいので，説明しておこう（印字例5）。

まず，TeXの記号は前回やったとおり，¥から始まるものが多くある。たとえば，

印字例5(70%に縮小)

8　　　　*CHAPTER 1.* これは章 (*CHAPTER*) です

1.2　記号に関する話

- バックスラッシュを書きたいときは、\ と書きます。このように、\ から始まる命令は過去にいくつか使いましたよね？
- fi や fl などは自動的に合字にします。
- したくないときには、fi こうしたり、fi こうします。後者のほうが少し広い間隔を空けます。
- -hyphen、–endash、—emdash など、
- 記号ではないし、日本人にはあんまり関係ないかもしれませんが…、Considering the history of bloodshed that religious differences have wrought elsewhere,Americans can be proud of the degree of religious tolerance which exists in their country. と、英語の長い文章が続いたときにも、禁則はきちんと行いますよね。
- 記号ではないし、日本人にはあんまり関係ないかもしれませんが……、Considering the history of bloodshed that religious differences have wrought elsewhere,Americans can be proud of the degree of religious tolerance which exists in their country. と、英語の長い文章が続いたときにも、禁則はきちんと行いますよね。
- 記号ではないし、日本人にはあんまり関係ないかもしれませんが…、Considering the history of blood-shed that religious differences have wrought elsewhere,Americans can be proud of the degree of religious tolerance which exists in their country. と、英語の長い文章が続いたときにも、禁則はきちんと行いますよね。
- 上の3つの違いがわかりましたか？

リスト2

```
29:
30: ¥section{記号に関する話}
31:
32: ¥begin{itemize}
33:  ¥item  バックスラッシュを書きたいときは、$¥backslash$と書きます。
34:         このように、$¥backslash$から始まる命令は過去にいくつか使いましたよね？
35:  ¥item  fiやflなどは自動的に合字にします。
36:  ¥item  したくないときには、f{}iこうしたり、f¥/iこうします。後者のほうが少し広い間隔を空けます。
37:  ¥item  -hyphen、--endash、---emdashなど、
38:  ¥item  記号ではないし、日本人にはあんまり関係ないかもしれませんが…、Considering the history o
f bloodshed that religious differences have wrought elsewhere,Americans can be p
roud of the degree of religious tolerance which exists in their country. と、英語の
長い文章が続いたときにも、禁則はきちんと行いますよね。
39:  ¥item  記号ではないし、日本人にはあんまり関係ないかもしれませんが……、Considering the history
of bloodshed that religious differences have wrought elsewhere,Americans can be
proud of the degree of religious tolerance which exists in their country. と、英語
の長い文章が続いたときにも、禁則はきちんと行いますよね。
40:  ¥item  記号ではないし、日本人にはあんまり関係ないかもしれませんが…、Considering the history o
f blood-shed that religious differences have wrought elsewhere,Americans can be
proud of the degree of religious tolerance which exists in their country. と、英語
の長い文章が続いたときにも、禁則はきちんと行いますよね。
41:  ¥item  上の3つの違いがわかりましたか？
42: ¥end{itemize}
```

印字例6(70%に縮小)

1.3　空白の命令 (横)

- 空白を制御します |　　　　|どうでしょう？ 10mm です。
- 空白を制御します |　　　　　　　　|どうでしょう？ 2cm です。
- 空白を制御します |　　　　|どうでしょう？ 0.5inch です。
- 空白を制御します | |どうでしょう？ 1point です。
- 空白を制御しまどうでしょう？ -1cm です。

文章の初めからスペースがほしい場合、hspace*を使います。
　　　　　　ここからです。

$¥backslash$

などはバックスラッシュを書く命令。わざわざ$~$で囲っているのは，この¥backslash命令は数式モードでしか使用できない命令だからである。これらはTeX本にも一覧があるし，ここで一覧を書き，いたずらにページを増やすのも無意味だと思うので，説明を避ける。

このほかにTeXではfiの合字などといった，いわゆるカーニングに関することがいくつかある。合字にしたくないときには，例(リスト2，36行目)にあるように{}や，¥/を間に挟めば，多少のスペースが空き，合字にはならない。

ハイフンやダッシュなどの指定も，-だけならハイフン，–と続けばen-dash，—となればem-dashになる。

記号とは関係ないが，禁則に関してもTeXは自動で行う。印字例5の英文を挟んだ3つの文章の違いがわかるかな？

あと，左上の部分にページ番号，右上の部分に章番号と章タイトルが入る。このような部分がいかにもbookという感じがする。

空白に関して

いままでレイアウトはすべてTeXに任せる思考できていたが，多少のレイアウト操作は空白をうまく利用して簡単にできるので，紹介しておこう。ここでは以前11月号で説明した¥;や¥:などといった命令とは違って，直接長さを指定するものについて説明する。

まずは横にスペースを空ける¥hspace{}命令(印字例6)。リスト3のように，10mm，2cm，0.5in，1pt，－1cmと具体的

リスト3

```
43:
44: ¥section{空白の命令(横)}
45:
46: ¥begin{itemize}
47:  ¥item  空白を制御します|¥hspace{10mm}|どうでしょう? 10mmです。
48:  ¥item  空白を制御します|¥hspace{2cm}|どうでしょう? 2cmです。
49:  ¥item  空白を制御します|¥hspace{.5in}|どうでしょう? 0.5inchです。
50:  ¥item  空白を制御します|¥hspace{1pt}|どうでしょう? 1pointです。
51:  ¥item  空白を制御します|¥hspace{-1cm}|どうでしょう? -1cmです。
52: ¥end{itemize}
53:
54: 文章の初めからスペースがほしい場合、hspace*を使います。
55:
56: ¥hspace*{3cm}ここからです。
57:
58: ¥newpage
```

な値で設定できる。単位は，mm，cm，inch，pointなどがあり，負の値を設定すると重ね打ちになる。inch，pointで設定する場合には注意が必要だ。なお，｛｝は全角。文の左端からスペースが必要な場合，¥hspace＊｛3cm｝というように＊を入れる。

縦の空白は，¥vspace{}を使う（印字例7）。¥vspace＊{3cm}とすれば，頭から3cm空けるということ。文章中に少し間隔を空けたいなら，¥vspace{1cm}とかすれば1cmの縦スペースが空く。¥hspaceと同じく，負の長さを指定すれば重ね打ちされる。

これら具体的な命令のほかに，少し縦に隙間を空ける¥smallskip，中ぐらいの¥midskip，大きく空ける¥bigskipなどがある。

ちなみに，印字例7の右上にあるのがページ番号で，左上にあるのがセクション番号とそのタイトルである。

印字例7（70%に縮小）

1.4. 空白の命令（縦）

1.4. 空白の命令（縦）

表用のスペースとして3 cm取りたい場合はこのようにします。こんな場所にきてしまうんですよね。
この下 1inch の部分に文章を書きたい場合、

と書きます。
負のスペースを入れてみました。
ここにきてしまいます。

リスト4

```
59:
60: ¥section{空白の命令(縦)}
61:
62: ¥vspace*{3cm}
63:
64: 表用のスペースとして3cm取りたい場合はこのようにします。こんな場所にきてしまうんですよ
ね。
65:
66: この下1inchの部分に文章を書きたい場合、
67:
68: ¥vspace{1in}
69:
70: と書きます。
71:
72: 負のスペースを入れてみました。
73:
74: ¥vspace{-3mm}
75:
76: ここにきてしまいます。
77:
78: ¥newpage
```

itemize環境とenumerate環境

itemize環境とenumerate環境は，箇条書きをするための命令。

まずは，itemize環境。文章を簡単に箇条書きにする。今回のソースの各所に使われているからわかるよね？

フォーマットは，

```
¥begin {itemize}
  ¥item 内容1
  ¥item 内容2
¥end {itemize}
```

印字例8（70%に縮小）

1.5 itemize 環境

1.5.1 標準的な itemize 環境の例

文章を印刷するとき……

- 細かくレイアウトを考えて文章を書くならば X-DTP。
- 簡単な文章を印刷するならシャーペン。
- 表や数式がたくさんあり、長めの文章を効率良く処理したいなら TeX。

とまあ、こんな感じです。

1.5.2 ラベルを変える場合の itemize 環境

こういうこともできます。

♡ 細かくレイアウトを考えて文章を書くならば X-DTP。

◇ 簡単な文章を印刷するならシャーペン。

♠ 表や数式がたくさんあり、長めの文章を効率良く処理したいなら TeX。

リスト5

```
79:
80: ¥section{itemize環境}
81:
82: ¥subsection{標準的なitemize環境の例}
83:
84: 文章を印刷するとき ¥ldots ¥ldots
85:
86: ¥begin{itemize}
87:   ¥item 細かくレイアウトを考えて文章を書くならばX-DTP。
88:   ¥item 簡単な文章を印刷するならシャーペン。
89:   ¥item 表や数式がたくさんあり、長めの文章を効率良く処理したいなら¥TeX 。
90: ¥end{itemize}
91:
92: とまあ、こんな感じです。
93:
94: ¥subsection{ラベルを変える場合のitemize環境}
95:
96: こういうこともできます。
97:
98: ¥begin{itemize}
99:   ¥item[$¥heartsuit$] 細かくレイアウトを考えて文章を書くならばX-DTP。
100:  ¥item[$¥diamondsuit$] 簡単な文章を印刷するならシャーペン。
101:  ¥item[$¥spadesuit$] 表や数式がたくさんあり、長めの文章を効率良く処理したいな
ら¥TeX 。
102: ¥end{itemize}
```

図1　瀧氏が使っているTeXの環境

というように書く。このitemize環境は普通，¥itemに相当するところに「・」($¥bullet $)がくると考えればよいのだが，これを別のモノに変える方法もある。印字例8の1.5.2のように，¥item[]と，[]の中にいろいろな文字を入れれば，それが「・」の代わりに利用されるというわけ。ジョークで，スートにしてみた。

itemize環境と同じような命令でenumerate環境がある。これはitemize環境の番号を自動的につけるものと考えればそれで当たり。よって，使い方はitemizeと同様。ただし，当然だが，itemize環境と違い，[ラベル]の指定はできない。

代わりといってはなんだが，enumerate環境は，多段ネストすることができる。これは，印字例9の1.6.2を参考にしてほしい。

まとめ

細かな指定をしない限り，これでだいたいのことはできるはずである。楽譜やベンゼン環とかを書かない限り，もう大丈夫だろう。

最初にいったとおり，私はまだまだTeXnicianではない。正直なところスタイルファイルは書いたことがないし，むしろ，TeXはアプリケーションとしてしか使っていないので，マニアックな使い方はできない。だから，アプリケーションとして簡単に使う方法を教えてきたつもりだが，いかがだったろうか？

なにせ，難しい図を挟みたい場合などは，¥vspace{}を酷使して，とりあえず場所を空けておき，別に図を印刷して張ったりして，TeXnicianからみたら，なんだそりゃって感じ。だから，私が教えられるのはこのぐらいなのだ。

話は変わって，左上のSX-WINDOW画面の印刷(図1)は，私がTeX入門講座を書くときのWINDOW風景。右上方にこの原稿，左上にTeXソース，右下に小さく開いたコンソール，そして左には1/2縮小して開いたsxpreviewの画面といったところ。なかなか便利である。12ドットフォントを使っていないのは，私が目が悪いから。20インチモニタでも1024×1024ドットになると16ドットじゃないときつい。

以上，麁枝大葉(そしたいよう)に4回分，原稿を書き連ねてきたが，読者の皆さんにこれを踏み台にして，TeXnicianになっていただければ幸いである。

印字例9(70%に縮小)

1.6　enumerate 環境

1.6.1　標準的な enumerate 環境の例

enumerate 環境の簡単な例です

1. 細かくレイアウトを考えて文章を書くならば X-DTP。
2. 簡単な文章を印刷するならシャーペン。
3. 表や数式がたくさんあり、長めの文章を効率良く処理したいなら TeX。

1.6.2　enumerate が複数ネストしていたときの例

1. 細かくレイアウトを考えて文章を書くならば X-DTP。
2. 簡単な文章を印刷するならシャーペン。
3. 表や数式がたくさんあり、長めの文章を効率良く処理したいなら TeX。
   (a) 細かくレイアウトを考えて文章を書くならば X-DTP。
   (b) 簡単な文章を印刷するならシャーペン。
   (c) 表や数式がたくさんあり、長めの文章を効率良く処理したいなら TeX。
      i. 細かくレイアウトを考えて文章を書くならば X-DTP。
      ii. 簡単な文章を印刷するならシャーペン。
      iii. 表や数式がたくさんあり、長めの文章を効率良く処理したいなら TeX。

リスト6

```
103:
104: ¥section{enumerate環境}
105:
106: ¥subsection{標準的なenumerate環境の例}
107:
108: enumerate環境の簡単な例です
109:
110: ¥begin{enumerate}
111:   ¥item 細かくレイアウトを考えて文章を書くならばX-DTP。
112:   ¥item 簡単な文章を印刷するならシャーペン。
113:   ¥item 表や数式がたくさんあり、長めの文章を効率良く処理したいなら¥TeX 。
114: ¥end{enumerate}
115:
116: ¥subsection{enumerateが複数ネストしていたときの例}
117:
118: ¥begin{enumerate}
119:   ¥item 細かくレイアウトを考えて文章を書くならばX-DTP。
120:   ¥item 簡単な文章を印刷するならシャーペン。
121:   ¥item 表や数式がたくさんあり、長めの文章を効率良く処理したいなら¥TeX 。
122:   ¥begin{enumerate}
123:    ¥item 細かくレイアウトを考えて文章を書くならばX-DTP。
124:    ¥item 簡単な文章を印刷するならシャーペン。
125:    ¥item 表や数式がたくさんあり、長めの文章を効率良く処理したいなら¥TeX 。
126:    ¥begin{enumerate}
127:     ¥item 細かくレイアウトを考えて文章を書くならばX-DTP。
128:     ¥item 簡単な文章を印刷するならシャーペン。
129:     ¥item 表や数式がたくさんあり、長めの文章を効率良く処理したいなら¥TeX 。
130:    ¥end{enumerate}
131:   ¥end{enumerate}
132: ¥end{enumerate}
133:
134: ¥end{document}
```

SIDE A
理論とゲームの間には……

Tan Akihiko 丹 明彦

基本的にゲームは離散的に進行していく
この離散的な進行が，シミュレーションを作るうえでどのような影響を与えるのか？
陥りがちな問題を取り上げ，解決方法を考えてみる

いわゆる次世代ゲーム機について

締め切りも近いというのにSEGA SATURN(以下サターン)とバーチャファイターを買ってしまった。前々から予約していたので，入手できるのはちょうど締め切りの頃であるため，遊ぶ時間を作れないのもわかっていたが，現在の状況は目の毒というか拷問に近い。この原稿を書き終わったら遊び倒すことにしよう。

で，このサターン版バーチャファイターであるが，実は私は買うまでサターンを見たことも触ったこともなかったので，初対面が製品版ということになる。ネットや噂などで伝え聞いていた話では，なぜか評価が二分していたのを不思議に思っていたのだが，現物を見て原因がわかった。単純に映像として見た場合，確かにサターン版はアーケード版に比べてグレードを落としている(ポリゴン数が減っているだけでなく，レンダリングの極端なミスも見受けられる)。しかし，ゲームとして総合的に捉えれば，映像としてのグレードダウンなど些細なことで，「ココロ」を完全に移植しきっているということなのだ。結局この評価の分かれ方は，その人にとってバーチャファイターがどんなゲームであったかを反映していると思うのだが，どうであろう。私自身は，映像としてのバーチャファイターをとても気に入っていたので，最初はサターン版の出来を物足りなく思ったのだが，2～3ゲームも遊んだら気にならなくなってしまった(もちろん自分でお金を払って買ったという事情も深く関係している)。

それにしても，このクラスの映像がたかだか4万円そこそこのハードウェアで実現されていることに，もう少し驚いたほうがいいのかもしれない。ゲーム機として見れば4万円が高いというのは事実だが，リアルタイム3次元CGのプラットフォームとして見れば，このコストパフォーマンスは驚異といっていいだろう。近年進境著しいDOS/Vマシンといえども，ポリゴン使用のアクションゲームという限定された状況のもとでは，現時点ではたった4万円の玩具にも及ばない(もっとも，1台のマシンに100万円ほどつぎ込む気があれば，量，質ともに上回ることができるのだが……)。ただ，不思議と次世代ゲーム機に共通しているのだが，できることに比べてあまりにも少ない主記憶は，近い将来(いやもうすでに？)重い足枷となってくるのだろう。

ゲーム制作におけるトラップ

無駄話が多いのは，ここ2カ月ほど私がその次世代ゲーム機方面にかかりきりになっていたせいである。非常にタイトなスケジュールをくぐり抜けてきた直後なので，少し調子が狂ってしまったようだ。

いまさらいうまでもないことだが，製品としてゲームを作るということは，本連載のように実験的なことをあれこれするのとは，同じようなプログラムを書いているように見えてもわけが違う。どんなに志の高い処理を行うプログラムを書いたとしても，バグが出ればなんの価値もない。ゲームソフト開発においては，研究などとは別次元のプログラミングが要求されるのである。私はいままでゲームソフトのレビュー記事もいくつか書いてきたし，辛辣なものもひとつや2つではなかったが，今回のような経験をすると，心情的にメーカーサイドに立ちたくなってしまう(もちろんそれは好ましいことではないのだが)。

さて，本連載の目指すところであるリアルタイムシミュレーションのための理論をゲームとして実装しようとした場合に，経験の足りないプログラマがはまるトラップがいくつかある。今回も申し訳ないが本筋のプログラミングについてはお休みをいただいて，そのへんを書いてみたい。なお，これはあくまで一般論で，決して私がそういうトラップに残ら

ずはまったわけではない。……ということにしておいていただきたい。

●前提(レーシングゲームの形態)

まず,最低限でも自車はプレイヤーの操作に応じてサーキットの中を走り続ける。たいていそこには,コンピュータあるいは別のプレイヤーが操作する他車がいて,速さを競い合い,あるいはバトルを繰り広げる。サーキットには道路などの走行可能な部分とグリーンゾーンやサンドトラップなどの走行可能であるが抵抗の大きい部分,それに壁などの走行不可能な部分とがある(図1)。

図1　一般的なレーシングゲームの形態

ここまではおおむね現実のレースでも成立する。コンピュータゲームに特有の状況として,自車あるいは他車は,ある時間間隔(毎秒30や60フレーム)で状態を更新しながら移動するというものがある。つまり,ゲーム世界においては時間は飛び飛びに流れるのである(これを離散的と呼ぶ)。むろん,精密に計算を行うことによって連続的な時間の流れをシミュレートすることも理論的には可能であるが,計算コストが大きい。十分に時間間隔が短かければ,動きも滑らかであり,近似的に連続とみなしても支障が出ないと考えられる。

●離散時間の副作用(壁のすり抜け)

時間が離散的に進むということは,車が飛び飛びに移動するということでもある。そこでは「すり抜け」という現象が起こる可能性が出てくる。

具体的に計算してみよう。たとえば時速300kmで走っている車を30フレーム毎秒の処理系で表現する場合,1フレームあたりの車の移動量は2.7mとなる(図2)。もっと遅い,たとえば10フレーム毎秒の処理系ともなれば1フレームで8.3mも進むことになってしまう。これは車の全長を大きく超える長さである。

この車が厚さ1mの壁に正面衝突すると,なんと車は壁に当たったとは判定されずにそのまま走り続ける。つまり壁をすり抜けるのである(図3)。これを「トンネル効果」(述語の用法はいい加減)とか呼んで冗談にしてしまうことは許されない。立派な「バグ」である。

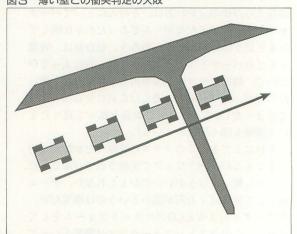
これを回避するにはどうするかといえば,まず正攻法として「連続時間で調べる」というのがあるが,これは場合によってはサーキットのデータ構造に抜

図2　離散時間の世界の車の動き

300km/hで走行する車を
30フレーム/秒の処理系で表示した場合

$$\frac{300\times10^3\,[m/hour]}{3600\,[sec/hour]\times30\,[frame/sec]}=2.7\,[m/frame]$$

10フレーム/秒の処理系で表示した場合

図3　薄い壁との衝突判定の失敗

本的な改良を施す必要性を生じる場合がある。

むろん計算コスト面でも不利である。あるいは，連続時間を近似するために「そのときだけより細かい時間間隔で調べる」という方法も考えられる。しかし，この場合手っとり早いのは「壁を厚くしてしまう」ことであろう。バカみたいだが，結構うまくいく。ゲームはあらゆるバグの発生を抑えなくてはならないが，逆にヤバそうなデータを先回りして片っ端から潰しておくという手もありなのである。

●他車との接触判定

レーシングゲームに限らずアクションゲームの類は，「ある瞬間の状態を表示する」→「次の瞬間の状態を求める」→「次の瞬間の状態を表示する」→……というサイクルで成り立っている。で，レーシングゲームにおいて自車/他車の動きの計算と接触判定の順序をいい加減にすると，妙な現象が起こる。

たとえば処理の順序を「自車の動き」→「接触判定」→「他車の動き」とすると，他車に接触していないように見えるのに接触したと判定されたり，またはその逆と判定されたりする。こうした現象を，実際に表示されない状態も含めて追いかけてみた様子を図4に示す。(A)は処理の順序を間違えたために実際には接触しないはずなのに接触したと判定されてしまう状況，(B)は正しい順序で処理した場合，(C)は処理の順序を間違えたために実際には接触しているのに接触しないと判定されてしまう状況，(D)は同じく正しい順序で処理した場合，をそれぞれ表している。

同様の現象はシューティングゲームなどでも起こるはずである。こういう現像はプレイヤーの感覚を裏切るので，非常にストレスがたまる。これを回避するには，処理の順序をきちんとするほかない。基本的な考え方は，自車も他車も同時に(並列に)動いているのだから，いったん全部動かして，移動後の状態を作っておいてから一斉に接触判定を行わなくてはならないということである。

小道具(折れ線グラフライブラリ)

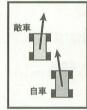
能書きばかりでもなんなので，少しはプログラミングのレベルで役に立つ話題も提供したい。ここで紹介するライブラリは，「折れ線グラフ」を実現する。もちろん自動車のシミュレーションを行うための小道具として開発したライブラリである。たとえばエンジンの出力曲線のように単純な式では表現できない特性を，簡単かつ軽い計算で実現することが可能になる（図5）。

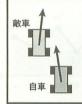
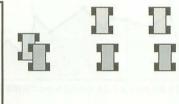
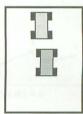
与えるデータは，表現したい特性曲線を要所要所でサンプリングした点列である(図6)。ありていにいえば，曲線の形がわかる程度に打った点である。

これを構造体LINEGRAPH型の変数として登録しておき，サンプリング点の間の値を補間して値を返す関数lineGraphGetValue()を呼ぶのである。折れ線グラフというくらいだから補間の方法は線形補間で，正確に曲線を表現することはできないが，サン

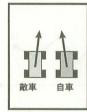
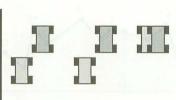
図4　他車との接触判定

(ケース1)　接触していないのに接触したと判定される

(A) 誤った処理順序

直前のフレーム　自車の動き計算　接触判定　他車の動き計算　次のフレーム（プレイヤーは見えない敵車に当たったように感じる）

(B) 正しい処理順序

直前のフレーム　自車の動き計算　他車の動き計算　接触判定　次のフレーム

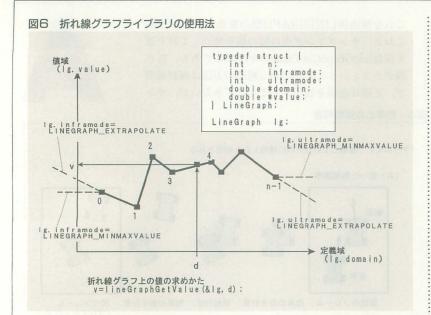
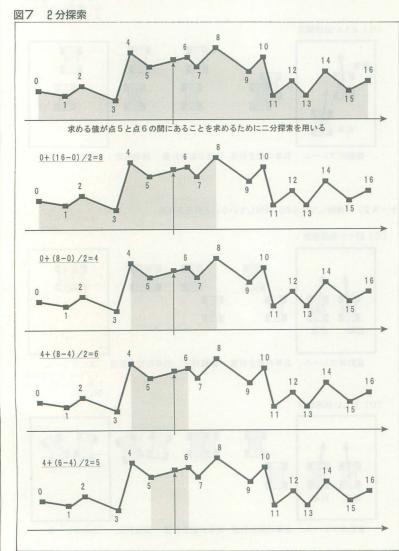
(ケース2)　接触しているのに接触していないと判定される

(C) 誤った処理順序

直前のフレーム　自車の動き計算　接触判定　他車の動き計算　次のフレーム（プレイヤーは敵車に粘りついたように感じる）

(D) 正しい処理順序

直前のフレーム　自車の動き計算　他車の動き計算　接触判定　次のフレーム

図6　折れ線グラフライブラリの使用法

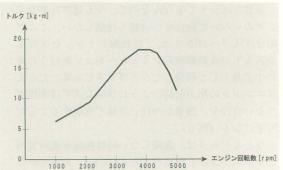
図7　2分探索

プリング間隔を十分細かく取れば，誤差は無視できる。

なお，サンプリング間隔は一定である必要はなく，サンプリング点の数にも制限は設けていない（補間アルゴリズムの性質上，極端に数が多いと速度が低下するかもしれない）。また，サンプリングした範囲外の値については，範囲よりも小さな部分(構造体メンバinframode)，大きな部分(同じくultramode)の各々について折れ線を外挿する(モードの値にLINEGRAPH_EXTRAPOLATEを与える)か，サンプリング範囲の端の値を用いる(同じくLINEGRAPH_MINMAXVALUE)かを設定することができる。

少々アルゴリズム解説的なこともしておこう。値の線形補間を行うためには，目的の値がどのサンプリング点の間にあるかを探らなくてはならない。しかも，ゲーム中に頻繁に用いることになるから，その探索はできる限り効率のいいものでなくてはならない。これらの要求を達成するために，2分探索(バイナリサーチ)を採用している。簡単にいえば，検索範囲を全範囲から始めて，目的の値が検索範囲の前半にあるか後半にあるかを調べた上で，検索範囲を半分に縮めて候補を絞っていく操作を繰り返す検索方法である(図7)。むろん，2分探索を使うのは折れ線グラフを定義している範囲内の場合であり，それ以外では外挿するなどの場合分けを行っている。

2分探索はサンプリングポイントの数が多い場合に特に速度面で効果が高い。たとえばサンプリングポイントが256個だった場合，単純に端から値を比較していく方式では平均128回の処理が必要だが，2分探索だとlog_2(256)＝8回ですむ。まあ全体の処理量から比べればゴミみたいなものだが，こういう小さな節約の積み重ねが全体のパフォーマンスを上げていくということもまた事実なのである。

リストは折れ線グラフライブラリのソースプログラム(Line Graph.c)とインクルードファイル(Line Graph.h)，それに簡単なサンプルプログラム(gtest.c)とそれをコンパイルするためのメイクファイル(Ma

図5　折れ線グラフライブラリが有用な例
(エンジン回転数ートルク特性曲線)

kefile)を添付しておく。次回以降，この折れ線グラフライブラリはいろいろな場面で使うことになるだろう。

終わりに

たったひとつでも完結した作品を作るといろいろなことが見えてくるものだ。最近はリッジレーサーにしろデイトナUSAにしろ，先人たちの作品をプレイしていても，車の挙動の端々に必然性のようなものが見えて，私の通った道を彼らはとっくの昔に通っていったのだと思わされるのである。まだ先は長いが，とりあえず追いつくところから始めなくてはならない，と決意を新たにしたところでまた次回。

■リスト1　LineGraph.c

```
 1: /*
 2:  *       LineGraph.c
 3:  *       - 折れ線グラフ
 4:  */
 5:
 6: #include         <stdio.h>
 7:
 8: #define          LINEGRAPHMAIN
 9: #include         "LineGraph.h"
10: #undef           LINEGRAPHMAIN
11:
12: /* 折れ線グラフ(g)の横軸の値(d)に対応する縦軸の値(戻り値)を得る */
13: double  lineGraphGetValue( LineGraph *g, double d )
14:
15:   int  n, n1, n2, nm;
16:
17:   n = g->n;
18:
19:   /* 定義域の端点に一致する場合 */
20:   if ( d == g->domain[0]   ) return ( g->value[0]   );  /* 定義域の
左端 */
21:   if ( d == g->domain[n-1] ) return ( g->value[n-1] );  /* 定義域の
右端 */
22:
23:   if ( d < g->domain[0] ) {  /* 定義域の左端より外 */
24:     /* 端点の値を用いる */
25:     if ( g->inframode == LINEGRAPH_MINMAXVALUE ) return ( g->valu
e[0] );
26:     /* 外挿した値を用いる */
27:     return ( g->value[1] + (g->value[0] - g->value[1])*(d - g->do
main[1])/(g->domain[0] - g->domain[1]) );
28:   } else if ( d > g->domain[g->n - 1] ) {  /* 定義域の右端より外 */
29:     /* 端点の値を用いる */
30:     if ( g->ultramode == LINEGRAPH_MINMAXVALUE ) return ( g->valu
e[n-1] );
31:     /* 外挿した値を用いる */
32:     return ( g->value[n-2] + (g->value[n-1] - g->value[n-2])*(d -
g->domain[n-2])/(g->domain[n-1] - g->domain[n-2]) );
33:   } else {  /* 定義域内 */
34:     /* どの点とどの点の間に入っているかを調べる(点数が多くなった場合に備えてバイナリ・サーチ
を用いている) */
35:     n1 = 0;     /* 初期検索範囲は定義域全体 */
36:     n2 = n-1;
37:     for (;;) {
38:       #ifdef  DEBUG
39:       /* 検索範囲を縮めていくうちに縮退してしまった場合
40:          (正常に動作している限りは起こりえない) */
41:       if ( n1 == n2 ) {
42:         fprintf( stderr, "***** n1 == n2 ¥n" );
43:         return ( g->value[n1] );
44:       }
45:       #endif
46:       /* 検索範囲を特定できた */
47:       if ( n2 - n1 == 1 ) break;
48:       /* 検索範囲を縮める */
49:       nm = ( n1 + n2 ) / 2;
50:       if ( d == g->domain[nm] )  /* 中央で一致した */
51:         return ( g->value[nm] );
52:       if ( d < g->domain[nm] ) {  /* 次の検索範囲は前半 */
53:         n2 = nm;
54:         continue;
55:       } else {        /* 次の検索範囲は後半 */
56:         n1 = nm;
57:         continue;
58:       }
59:     }
60:     /* 隣接する点の間での内挿により値を求める */
61:     return ( ((d - g->domain[n1])*g->value[n2] + (g->domain[n2] -
d)*g->value[n1])/(g->domain[n2] - g->domain[n1]) );
62:   }
63:   /*NOTREACHED*/
64: }
```

■リスト2　LineGraph.h

```
 1: /*
 2:  *       LineGraph.h
 3:  *       - 折れ線グラフ
 4:  */
 5:
 6: #ifndef __LINEGRAPH_H__
 7: #define __LINEGRAPH_H__
 8:
 9: typedef struct  _LineGraph {
10:                /* 折れ線グラフの点の数 */
11:         int      n;
12:                /* 定義域外の処理方式(端値/外挿) */
13:         int      inframode;
14:                /* 定義域外の処理方式(端値/外挿) */
15:         int      ultramode;
16:                /* 定義域(配列のサイズはnで昇順にソートされていること) */
17:         double  *domain;
18:                /* 値域(配列のサイズはn) */
19:         double  *value;
20: } LineGraph;
21:
22: #define LINEGRAPH_MINMAXVALUE   0        /* 定義域外は端値を用いる */
23: #define LINEGRAPH_EXTRAPOLATE   1        /* 定義域外は外挿する */
24:
25: #ifndef LINEGRAPHMAIN
26: double  lineGraphGetValue( LineGraph *g, double d );
27: #endif  /* LINEGRAPHMAIN */
28:
29: #endif  /* __LINEGRAPH_H__ */
```

■リスト3　gtest.c

```
 1: /* 折れ線グラフのテスト */
 2:
 3: #include          "LineGraph.h"
 4:
 5: LineGraph         lg;
 6: double            lg_domain[] = {-21,-10,-2,3,10,12,30};
 7: double            lg_value[]  = {22,20,18,18,25,0,13};
 8:
 9: void    draw_lg()
10: {
11:   int  i, j, k, f;
12:   double  v;
13:   for ( i = -40; i <= 40; i++ ) {
14:     v = lineGraphGetValue( &lg, (double)i );
15:     printf( "%+03d: %5.1f ", i, v );
16:     f = 0;
17:     for ( k = 0; k < lg.n; k++ ) if ( (double)i == lg.domain[k] )
f = 1;
18:     if ( f == 1 )
19:       for ( j = 0; j <= (int)v; j++ ) printf( "*" );
20:     else
21:       for ( j = 0; j <= (int)v; j++ ) printf( "." );
22:     printf( "¥n" );
23:   }
24: }
25:
26: void  main()
27: {
28:   lg.n = 7;
29:   lg.domain = lg_domain;
30:   lg.value = lg_value;
31:
32:   printf( "¥n¥n¥n¥n端点の値を用いる¥n¥n" );
33:
34:   lg.inframode = LINEGRAPH_MINMAXVALUE;
35:   lg.ultramode = LINEGRAPH_MINMAXVALUE;
36:
37:   draw_lg();
38:
39:   printf( "¥n¥n¥n¥n外挿する¥n¥n" );
40:
41:   lg.inframode = LINEGRAPH_EXTRAPOLATE;
42:   lg.ultramode = LINEGRAPH_EXTRAPOLATE;
43:
44:   draw_lg();
45: }
```

■リスト4　MakeFile

```
 1: all: gtest.x
 2:
 3: %.o: %.c
 4:         gcc -O -c $<
 5:
 6: gtest.x: gtest.o LineGraph.o
 7:         gcc -o gtest.x gtest.o LineGraph.o
 8:
 9: gtest.o: gtest.c LineGraph.h
10: LineGraph.o: LineGraph.c LineGraph.h
11:
12: clean:
13:         rm -f LineGraph.o gtest.o gtest.x
```

(で)のショートプロぱーてぃ —— その64

今年こそ我に明かりを!

Komura Satoshi 古村 聡

冬の真っ只中，風などひかず元気にお過ごしでしょうか？ 今月のプログラムはゲームが2本にユーティリティが1本です。どのプログラムも何か懐かしさを思い起こさせます。たまには，しんみりと楽しんでみるのもいいかもしれませんね。

illustration：T.Takahashi

みなさま，クリスマスや大掃除にお正月といろいろなことで忙しい年末年始ですが，いかがお過ごしでしょうか？

私どもライターは，年末ともなると年末進行で，馬車馬のように働かなきゃいかんのだ，ばっきゃろー。だから，クリスマスなんてありゃしない。

世間じゃ街が，もみの木が，ただ立ってるだけの電信柱でさえ電飾つけてキンキラキンだってーのに……あたしゃ電信柱以下の存在なのか？

ううっ，街ですれ違うみんなは浮かれているというのに，私は原稿を書かないと編集室に行っても編集さんに怒られてしまうのね……ってほとんどマッチ売りの少女のシチュエーションだぞ，これは。

うううう，私にも明かりを〜。

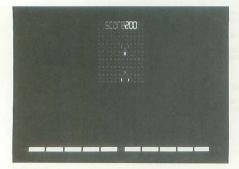
明かりが欲しい

では今月の1本目のプログラムです。

キーボードLED明るさ調節プログラム「あかりちゃん」です。どうぞっ。

AKARI.X for X680x0

(要アセンブラ，リンカ)

大阪府　牛島真一

REDLED.C

これはキーボードのLEDの明るさを調節するためのプログラムです。

プログラムはアセンブラのソースリストの形で掲載されています。まず，Human68k標準添付のエディタであれば，

A>ED AKARI.S

と，これから作るファイル名が「AKARI.S」という名前になるようにエディタを起動してください。それからリスト1をエディタで打ち込んで，

A>AS AKARI.S

A>LK AKARI.O

とアセンブル，リンク作業をすれば実行ファイルAKARI.Xを作ることができます。

これは，キーボードの［INS］や［全角］［ローマ字］といったキーについているLEDの明るさを，4段階に調節するプログラムです。いつもCAPSキーをピカピカに光らせていると，なんだかかわいそうな気がするでしょ（じゃあ消せよって？）。

で，使い方なんですけど，

A>AKARI /明るさ（0〜3）

という形で使います。明るさは0〜3までの数値。ふだんのX68000のLEDの明るさが3に相当します。ただし，このプログラム，引数の数値は1文字分しか見ていないから気をつけてくださいね。つまり，

A>AKARI /10

なんて打った場合には，プログラムはなにごともなかったかのように終了するので「あれっ？ もしかしてこれが10の明るさなのかな？」と思ってしまいますが，設定は1の“やや暗い”になっています。

後ろになにもない場合や，間違ったスイッチを指定した場合はヘルプメッセージを表示して終了します。また，スイッチを2つつけた場合もエラーを出さずに終了します（初めに現れたスイッチを実行して，それ以降を無視してるんですね）。

あぁ，これで私にもやっと明かりが……って，このプログラムは明かりを暗くするプログラムじゃないかっ！

ううっ，明かりが欲しいと思ったらどんどん暗くなっていくなんて，まさしくマッチ売りの少女のようだわ。このまま天に召されていくのね……（おい)。このプログラムはX68000のハードウェアを直接叩いているのですね。そうか，こんな機能もあったのか，X68000には。ふーん。

作者の牛島さんはこのプログラムをフリーソフトにするそうです。

そうそう，このプログラムを実行すると，リセットしても電源を切っても，またX68000を使うときには前に設定したLEDの明るさがしっかり残っています。だから，「なんか，LEDが暗いんですけど」ってX68000を修理に出すようなことがないように気をつけましょう。

古き良き時代が浮かんで……

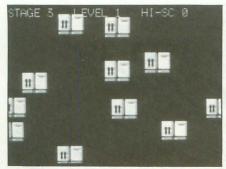
さてさて，2本目のプログラムはちょっと小粋でイナセな電卓風ゲーム。毎度お馴染み平井さんの作品でREDLED.Cです。

REDLED.C for X680x0

(要Cコンパイラ)

三重県　平井栄治

ちょっとレトロな感覚のシューティングゲームです。

このプログラムはCコンパイラのソースリストの形で書かれています。エディタでリスト2を入力，セーブしてから，

DOT.BAS

A＞CC /Y /W REDLED.C

でコンパイルして実行ファイルを作ってください。XCでなく，gccを使う場合はバージョンによってはコンパイラの問題でうまく動かないことがあるようです。Oh!X付録ディスクに添付されていたGnuCなどを使う場合には，

A＞gcc redled.c iocslib.a baslib.a

のように最適化をしないでコンパイルを行ってください。で，遊び方は，

A＞REDLED

と起動するだけです。

ここで“s or esc”と表示されているので，Sキーを押すとゲームスタート！

マイキャラをカーソルキーの［←］［→］で動かし，敵の弾に当たらないように操って，自弾を敵機（20点）に当ててください。敵機が出つくしたり（50機），自機に敵機や敵の弾が当たるとゲーム終了です。ESCキーを押すと親プロセスに実行を移します。そうそう，自分の弾はスペースキーで出ます。

いや～っ，懐かしい。あのねあのね，攻めてくる敵機がね。画面の上のほうで右左に水平移動したあとで，シューッと斜めに降りてくるんですよぉ。なんか，昔懐かしのLSIゲーム，ギャ○ク○アンみたいではないですか。考えてみたら，LSIゲームが出たのってもう15年くらい前ですもんね。あの頃はこの手のゲーム買ってもらえなくてくやしくてさ～，ってどーして今月はそういうびんぼうくさい話になるのだ，まったくもうっ（苦笑）。

や～，でもこれ本当に敵のアルゴリズムがよくできてますよね。ときどき下から上に斜め移動する敵もいるんだけど，こいつはいきなり自機X座標－2～自機X座標＋2の位置には出現しないようにプログラムされているんです。基本的に敵機は，敵弾発射可能時には自機のいるほうに，敵弾発射不可時には自機のいないほうに移動するようになっているし。さすが常連の平井さん。やりますね。とてもCのプログラムは初めてとは思えない。

ちなみに私はシューティングゲームが苦手なんで（ちなみに最近一番得意なゲームは「プリティセーラー18禁」だったりします。しくしく……）十分なくらいむずいと思うんですが，簡単すぎるぞ，と思う方は，

A＞REDLED /A

のようになにかひとつ，適当なオプションをつけてください。

それから（あんまりいないと思うけど）画面を256×256ドットモードで使っている場合は，

A＞REDLED /A /A

のように，2つ適当なオプションをつけてください。

要するに，コマンドラインのコマンド名を除いた文字列数の0ビットが立っていると敵の速度が増し，1ビットが立っていると表示位置が左端になります（しかし，誰がこんなスピードでできるんだ……できるやつは人間じゃねぇっ）。

そうそう，このプログラムは実行中，テキストパレットを変更しています。つまり，REDLED実行前のカラーコードとREDLED用のカラーコードをきちんと交換しているのです。ですから，標準値以外の，たとえば「電卓は灰色の地にセピア色で使っているのさ……ふっ」という心のさびしい方も，無理やり白と黄色の明るいカラーにさせられることはありません。安心してダークな気分で遊びましょう。どろどろどろ（いかんっ，どうしても今月は原稿がダークな方向に行ってしまう）。

走馬灯のように

それでは今月最後のプログラムを紹介しましょう。3本目のプログラムもゲームプログラムです。千葉県の森川さんの作品で，なんだか，昔が無性に懐かしいショートゲーム，DOT.BASです。どうぞっ。

DOT.BAS for X680x0

（要CコンパイラPRO-68K）

千葉県　森川忠敬

このプログラムはBASICのリストの形で書かれていますが，コンパイラ専用です。まず，Cコンパイラのパッケージと，

SCRM.PCM

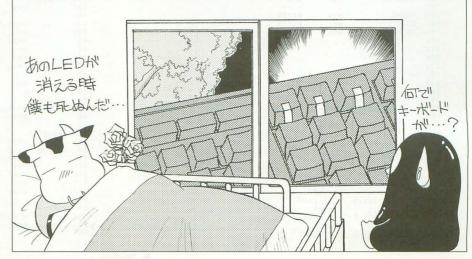
APL.PCM

という名前でPCMのファイルを2つ用意してください。

この2つは面クリアに失敗したときと成功したときの効果音として必要ですので，適当にZ-MUSICのディスクからコピーして名前を変えるなり，自分でなにか録音するなりして好きに用意してくださいね。

で，まずリスト3のアセンブラのリストDOT_LIB.Sを作ってください。それからリスト4のBASICのリストを打ち込みます。こちらのファイル名はDOT.BASです。両方正しく打ち込んでディスクに保存できたら，この両方をいっしょにコンパイルします。Cコンパイラを使って，

A＞CC DOT.BAS DOT_LIB.S

としてコンパイルしてください。正しくDOT.Xというファイルが作れましたか？

できたら，遊び方です。

このゲームは迫りくるロクハチ（仮名）をひたすら避けるのだ！　というドットよけゲーです。ある程度進むとステージクリアとなり，いくらか大きくなったロクハチが再び向かってきます。ロクハチに当たってしまった，もしくは画面の上下からはみ出したらゲームオーバーです。

このゲームはジョイスティックで操作し，Aボタンで上昇，左右は左右，上下でスピードが2変速でシフトチェンジできます。Aボタンを押さないとドットは重力に従って落ちていきます（宇宙空間で重力が働くか？　などという突っ込みはなしね）。

また，タイトル中にゲームのレベルとウエイトの有無が設定できます。レベルは1～9までで，出現するロクハチの量を決定します。ウエイトは，オンのとき標準，オフのとき倍速でゲームができます。Bボタンで終了します。

うーむ，これもまた懐かしいタイプのゲームが～……横スクロールゲームですねっ。そういえば，こんなのを私もショートプロの連載が始まる前（ってことは6年前か）に作ったような記憶が……当時はX1だったけど（1989年3月号参照）。もしかして森川さん，適当にX68000の絵を描いてみて，動かしてたらゲームになっちゃった，とかいうんじゃないですよねー。いや，今月はレトロだな～。いかんっ，年の暮れにひとり寂しく明かりをつけて走馬灯のように昔のことばっかり思い出すなんてまるで……(以下略)……ああ，やっぱり今月はこんな話題ばっかりだ。

でも，適当な絵を「へろへろ～っ」と描いてそいつを動かしてみて，あ，ゲームができちゃうというのはショートプロの作り方の王道でありますから，よしとしましょう。それに，ちゃーんと，X68000が大きくなったりする工夫も見られますしね（結構インパクトでかいよね，これ）。

そうそう。ゲームには直接関係ないですが，PCM8が常駐しているとスクロールがスムーズにならない場合があります。そういうときは常駐物を外してからゲームをしましょう。このプログラムには，音楽関係のドライバ（AD PCMを再生するため）とFLOATn.X以外には特に必要となるドライバはありません。

さて，今月はこれにておしまいっ。あわれな（で）は天に召されていきました……な，わけないでしょうっ！　今月は正月だし，これから節分，ひなまつりと季節は進んでいくのであります。ああ，輝ける和風の季節。へへーんだ，年末のクリスマスの仲間なんかあとはハロウィングらいしかないじゃないか。悔しかったら，この俺様にお年玉を与えてみよ！　ってことでまた来月。

リスト1　AKARI.S

```
 1: *キーボードLED明るさ調節プログラム「あかりちゃん」
 2: *                      akari.x
 3: *                              94/10/6  Programed by S.牛島
 4:
 5: _B_SUPER        equ     $81
 6:
 7: _PRINT          equ     $ff09
 8: _EXIT           equ     $ff00
 9: _EXIT2          equ     $ff4c
10:
11:         .text
12:         .even
13:
14: ent:
15:         lea.l   mysp,sp                 * spの初期化
16:
17:         tst.b   (a2)+
18:         beq     usage
19:
20:         bsr     chkswt
21:
22:         .dc.w   _EXIT                   * おしまい。チャンチャン
23:
24: chkswt:
25:         bsr     skipswt
26:
27:         cmpi.b  #'0',(a2)
28:         beq     black0
29:         cmpi.b  #'1',(a2)
30:         beq     black1
31:         cmpi.b  #'2',(a2)
32:         beq     bright0
33:         cmpi.b  #'3',(a2)
34:         beq     bright1
35:
36:         bra     usage
37:
38: skipswt:
39:         bsr     skipsp
40:
41:         cmpi.b  #'/',(a2)
42:         beq     relate
43:         cmpi.b  #'-',(a2)
44:         beq     relate
45:         cmpi.b  #'+',(a2)
46:         beq     relate
47:
48:         bra     usage
49:
50: relate:
51:         addq.l  #1,a2
52:         rts
53:
54: skpsp0:
55:         addq.l  #1,a2
56: skipsp:
57:         cmpi.b  #$20,(a2)               * スペースならスキップ
58:         beq     skpsp0
59:         cmpi.b  #$9,(a2)                * タブならスキップ
60:         beq     skpsp0
61:         rts
62:
63: black0:
64:         move.b  #%01010111,d1           * '暗い'
65:         bra     variety
66:
```

▶「料理の鉄人」のテーマ曲のCDが出てたら誰か教えてください。　舌 孝(22)京都府

```
 67: black1:
 68:         move.b  #%01010110,d1   * 'やや暗い'
 69:         bra     variety
 70:
 71: bright0:
 72:         move.b  #%01010101,d1   * 'やや明るい'
 73:         bra     variety
 74:
 75: bright1:
 76:         move.b  #%01010100,d1   * '明るい' (初期設定値)
 77:
 78: variety:
 79:         movea.l #0,a1
 80:         moveq.l #_B_SUPER,d0
 81:         trap    #15
 82:
 83:         movea.l d0,a1
 84:
 85:         move.b  d1,$e8802f      * MFP USARTﾃﾞｰﾀﾚｼﾞｽﾀ
 86:
 87:         moveq.l #_B_SUPER,d0
 88:         trap    #15
 89:
 90:         bsr     tune            * 何に、設定したか表示する
 91:
 92:         rts
 93:
 94: tune:
 95:         lea.l   resmes,a0
 96: tune1:
 97:         cmpi.b  #'0',(a0)+
 98:         bne     tune1
 99:
100:         move.b  (a2),-(a0)
101:
102:         pea.l   resmes
103:         .dc.w   _PRINT
104:         addq.l  #4,sp
105:
106:         rts
107:
108: usage:
109:         pea.l   hlpmes          * ﾍﾙﾌﾟを表示して
110:         .dc.w   _PRINT
111:         addq.l  #4,sp
112:
113:         move.w  #-1,-(sp)       * EXITｺｰﾄﾞ -1 で終了
114:         .dc.w   _EXIT2
115:
116:         .data
117:         .even
118:
119: hlpmes:
120:         .dc.b   '-Akari.X- ｷｰﾎﾞｰﾄﾞLED明るさ調節「あかりちゃん」P
rogramed 1994/Oct By S.牛島',13,10
121:         .dc.b   '[ 使い方 ]: akari  /(switch)',13,10
122:         .dc.b   '    n     : 明るさ指定 (0～3)',13,10
123:         .dc.b   '[ちゅうい]: 調節したLED明るさは、ﾘｾｯﾄしても元に戻りま
せん。',13,10
124:         .dc.b   '              再度設定して下さい。[初期値3]',13,
10,00
125:
126: resmes:
127:         .dc.b   '§ LED明るさを、0 に調節しました §',13,10,00
128:
129:         .stack
130:         .even
131:
132: mystack:
133:         .ds.l   256
134: mysp:
135:         .end
```

リスト2 REDLED.C

```
  1:          int                                  g,h=0xdfad80;
  2:    short int                         *i,j=0,k,m=0,n[532];
  3: struct o{ int                                           o;
  4:    short int                                           p;
  5:          }                                         p[59];
  6:       char *q,r=0,s=0,t=5,u=5,v=5,w=5,x[1158],y[14],
  7:    z[22]={119, 36, 93,109, 46,107,123, 37,127,111, 83,
  8:          19, 95,  8, 20,  0, 44,108, 32, 13, 10,  0}
  9:   ;void a(int b){int c,d,e,f;switch(b){case 0:m=579-m;
 10:           for(c=0;c<288;   ){  b=4-b;x[c+++m]=6<<b;}
 11:           for(   ;c<576;   ){          x[c+++m]=   0;}
 12: break;case 1:for(   ;b< 15;b++){q=&z[19];c=B_PRINT(q);}
 13:     h+=c<<11;i=h-  7552;for(c=     0;c<       8;c++){
 14:                         for(d=     0;d<      12;d++){
 15:                         for(e=   224;e<     252;   ){
 16:       *i=n[e++];i+=64;}    i-=1791;}i+= 1780;}
 17:       b= 0;i=h-  7548;for(c=     0;c<       2;c++){
 18:                         for(d=     0;d<       5;d++){
 19:                         for(e=p[b].p;e<p[b].p+28;   ){
 20:       *i=n[e++];i+=64;}b++;i-=1791;}i+=65531;}break;
 21:   case 2:b=51;i=h+127108;for(d=     0;d<       8;d++){
 22:                         for(e=p[b].p;e<p[b].p+28;   ){
 23:      .*i=n[e++];i+=64;}b++;i-=1791;}              break;
 24:   case 3:     i=h+127108;for(d=     0;d<       8;d++){
 25:                         for(e=   420;e<     448;   ){
 26:       *i=n[e++];i+=64;}    i-=1791;}              break;
 27:   case 4:          a(0);for(b=1;b<14;b++){
 28:                        for(c=0;c< 2;c++){if(-1<y[2+b]){
 29:    d =22*c+m+y[b]*4+y[1+b]* 48;x[  3+d]=0;x[291+d]= 6;
 30:    d+= 4*c                     ;x[    d]=0;x[288+d]=96;
 31:                                }else{}b+=3;}}b=j;c=576;
 32:    d =     m+   t*4+        240;x[315+d]=6;x[312+d]=96;
 33:    d =     m+   u*4+      v* 48;x[289+d]=6;x[290+d]=96;
 34:         for(d=100;d> 0;d/=10){e=b/d;b-=d*e;x[c+++m]=e;}
 35:         for(     ;d<51;     ){              p[d++].o=0;}
 36:         b= 0;for(c=0;c<576;c++){if(x[m+c]!=x[579-m+c]){
 37:            d=c<288?0:1;e=c-d*288;f=e/24;p[b].p=x[m+c];
 38:        p[b++].o=(d<<17)+e-(f*3<<3)+(((f>>1)+f*3)<<9)+h;
 39: }else{}}b=48;for(   ;c<579;c++){if(x[m+c]!=x[579-m+c]){
 40:                                           p[b].p=x[m+c];
 41:        p[b++].o=                          122382+c+c+h;
 42: }else{}}         break;case 5:q=0xe88001;while(0==16&*q);
 43:                  break;case 6:q=0xe88001;while(   16&*q);
 44:                  break;case 7:          for(b= 0;b<48;   ){
 45:  q=p[b].o;        if(q){c=   p[b++].p;for(d= 0;d<10;d++){
 46: *q=       c;q+=128;}}else{break;}}     for(b=48;b<51;   ){
 47:  i=p[b].o;        if(i){c=28*p[b++].p;for(d= 0;d<28;d++){
 48: *i=n[c++];i+= 64;}}else{break;}}      break;case 8:a(5);
 49: a(6);for(b=0;b<4;){c=b+51;d=p[c].o;p[c].o=TPALET(b,-1);
 50:            TPALET(b++,d);}break;case 9:if(  2&BITSNS(0)){
 51:   r=2;}else                             if(128&BITSNS(3)){
 52:   r=1;                               while(128&BITSNS(3)){
 53:       }}else{}}}void main(int b){int c,d,e,f;B_SUPER(0);
 54:        g=1&--b?12:16;h=2&b?h:h+36;b=0;for(c=0;c<19;c++){
 55:        n[b++]=0;                          for(d=0;d< 7;d++){
 56:        switch(d){case 0:case 3:case 6:     e=0;   f= 8184;
 57:                  break;case 2:case 5:             f=     6;
 58:                  break;case 1:case 4:      e=0;   f=24576;
 59:                  }                           e=1&z[c]>>d?f+e:e;
 60:        switch(d){case 0:case 3:case 6:for(f=0;f< 2;f++){
 61:        n[b++]=e;} break;case 2:case 5:for(f=0;f<10;f++){
 62:        n[b++]=e;}   p[ 0].p=392;p[ 2].p=364;p[ 4].p=504;
 63:       }}n[b++]=0;}   p[ 1].p=448;p[ 3].p=476;p[ 5].p=140;
 64:        p[51].o=     0;p[51].p=140;p[55].p=420;p[ 6].p=280;
 65:        p[52].o=16912;p[52].p=420;p[56].p=336;p[ 7].p=   0;
 66:        p[53].o= 1984;p[53].p=   0;p[57].p=140;p[ 8].p=308;
 67:        p[54].o=65534;p[54].p=308;p[58].p=280;p[ 9].p=336;
 68: for(b=1155;b<1158;){x[b++]=11;}
 69:           a(0);a(1);y[3]=y[6]=y[10]=y[13]=-1;while(r<2){
 70: for(b=   4; 9-r>b;){   a(b++);}j=r=0;k=g*3;u=t;v=5;w=5;
 71:                 a(2);y[3]=y[6]=y[10]=y[13]=-1;while(r<1){
 72:                                 a(9);            }while(r<2){
 73:                     a(3);a(6);r=0;break;}while(r<1){
 74:                     a(5);a(9);                while(r<1){
 75: b=40&BITSNS(7);if(b==8){b=-1;}else if(b==32){b=1;}else{
 76: b=             0;}                if(s!=b                ){
 77:       s=b;t+=s;if(t==6){t= 5;}else if(t==-1){t=0;}else{
 78:          }u=w==5?t:u;}else{}if(5==w&&       32&BITSNS(6)){
 79: v--;w--;}else{}w=v<5?--w:w;if(   w){                 }else{
 80: v--;w=4;}                         if(0 >v){u=t;v=5;w= 5;}else{
 81:          }    for(                             c=0;c<14;c+=7){
 82:               if(                                   -1==y[3+c]){
 83:          }else{if(                                    --y[3+c]){
 84:               if(y[6+c]==g/2+1&&t==y[1+c]&&g/2==y[3+c]){
 85:          ++y[5+c];y[6+c]--;}else{                    }}else{
 86:  y[2+c]+=y[  c];y[3+c]=g;       if(y[2+c]>-1&&y[2+c]<6){
 87:                                  if(y[1+c]> 0&&y[1+c]<5){
 88:      d=g/2+1==y[6+c]?1:-1;      if(           y[1+c]>t){
 89:      d=                  -d;}else if(           y[1+c]<t){
 90:                              }else if(                 3>t){
 91:      d=                  -d;}else{                          }
 92:                 y[1+c]+=d;}else if(           y[1+c]   ){
 93:                 y[1+c]-- ;}else{              y[1+c]++;}
 94:    if(g/2+1==y[6+c]){y[4+c]=y[1+c];y[5+c]=y[2+c];}else{}
 95:     }else{y[3+c]=-1;y[6+c]=g/2+1==y[6+c] ?-1 :  y[6+c];}
 96:                 }}          if(g/2+1==y[6+c]||-1==  y[6+c]){
 97:                 }else{   if(                         --y[6+c]){
 98:                 }else{   if(                      6!=++y[5+c]){
 99: y[6+c]= g/2  ;}else     if(    0==y[  c]&&-1!=  y[3+c]){
100: y[6+c]= g/2+1;                      y[4+c] =     y[1+c] ;
101: y[5+c]=y[2+c];}else{                y[6+c] =          -1 ;
102:                 }}}}     if(                     k &     63){
103:            --k;}else{  for(                    c=0;c<14;c+=7){
104:                          if(   -1==y[3+c]&&-1==  y[6+c]){
105:          break;}else{}} if(                      c==     14){
106:            ++k;}else{   if(                      k <   3200){
107: y[3+c]= g     ;y[c] =rand()%3;y[2+c]=y[5+c]=0==y[c]?5:0;
108: y[6+c]= g/2+1;   k+=  g*3+64;             switch(y[c]-- ){
109:           case 0:if(t>2){y[1+c]=y[4+c]=     rand()%(t-2);
110:                          }else{y[1+c]=y[4+c]=t+3+rand()%(3-t);
111:    }break;case 1:             y[1+c]=y[4+c]=     rand()%    6;
112:     break;case 2:                          d=t-2+rand()%    5;
113:                          if(d<0){d=0;}else if(5<d){d=5;}else{}
114:                              y[1+c]=y[4+c]=                d;
115:                          }}else{k++;}}}b=0;for(c=0;c<14;c+=7){
116:           if(v<5&&u==y[1+c]&& v==y[2+c]&&-1!=y[3+c]    ){
117:      b=1;j+=20;y[3+c]=-1;y[6+c]=g/2+1==y[6+c]?-1:y[6+c];
118:          }else{}r=t==y[1+c]&& 5==y[2+c]&&-1!=y[3+c]?1:r;
119:                   r=t==y[4+c]&& 5==y[5+c]&&-1!=y[6+c]?1:r;
120:                 }r=k >  3199&&-1==y[  3]&&-1==y[ 10]?1:r;
121:  if(b){u=t;v=5;w=5;}else{}j=j>999?999:j;a(4);a(6);a(7);
122:          break;}}}a(8);while(B_KEYSNS()){B_KEYINP();}}/*
123: REDLED              C           6968  94-06-18  12:00:00*/
```

▶ACEが故障し，PROを買ったのだが，拡張I/Oスロットの中古よりも，増設フロッピーディスクの中古よりも安いのには，嬉しくもあり悲しくもあった。原田　健史(27)東京都

リスト3 DOT_LIB.S

```
 1:          .include        doscall.mac
 2:          .include        iocscall.mac
 3:
 4:          .globl  _dotwait
 5:          .globl  _dotprio
 6:
 7:          .text
 8:
 9: _dotwait:
10:          clr.l   -(sp)
11:          DOS     _SUPER
12:          addq.l  #4,sp
13: loop:
14:          btst.b  #4,$E88001
15:          bne     loop
16:          move.l  d0,-(sp)
17:          DOS     _SUPER
18:          addq.l  #4,sp
19:          rts
20:
21: _dotprio:
22:          move.b  #$21,d1
23:          lea     $E82500,a1
24:          IOCS    _B_BPOKE
25:          rts
26:
27:          .end
```

リスト4 DOT.BAS

```
  10 screen 0,0,1,1:window(0,0,1023,1023):dotprio()
  20 color 1:color[,41982]
  30 int x,y,st,tr,spd,hx,hy,bx1,bx2,p,dm
  40 int sc,x68,wa,ro=1,lv=1,hsc(9)
  50 float ex,ey,ma
  60 dim char gr1(255),gr2(32767)
  70 dim char pcm1(19167),pcm2(62110)
  80 sound():sprite():enemy()
  90 /****************
 100 while 1
 110   title()
 120   x=50:y=128:spd=1:sc=0:ma=1:hx=0:hy=0:p=0:ro=1
 130   stage_set()
 140   repeat
 150     st=stick(1):tr=strig(1)
 160     x=x+((st=6)-(st=4))*spd
 170     y=y+((tr=0 or tr=2)-(tr=1 or tr=3))*spd
 180     if st=8 or st=2 then spd=(st=8)+(st=2)*2
 190     hx=(hx+1) mod 768
 200     sc=sc+1
 210     if hx=0 then {
 220       hy=(hy+256) mod 1024
 230       if hy=0 then { ma=ma+0.5#:ro=ro+1
 240         a_play(pcm2,4,3)
 250         dotwait():home(0,0,0)
 260         for i=0 to 3
 270           symbol(65-i,110,"CLEAR!",1,1,2,7+i,0)
 280         next
 290         stage_set()
 300       }
 310     }
 320     p=point(hx+x-16,hy+y-16)
 330     if (x<16 or x>271 or y<16 or y>271) then p=1
 340     if wa=0 then for i=0 to 1000:next
 350     dotwait():home(0,hx,hy)
 360     sp_set(0,x,y,260):bgscr()
 370   until p<>0
 380   a_end():a_play(pcm1,4,3)
 390   for i=0 to 3
 400     symbol(hx+25-i,hy+90,"GAME OVER",1,1,2,7+i,0)
 410     symbol(hx+60-i,hy+170,"SCORE "+str$(sc),1,1,2,7+i,0)
 420   next
 430   for i=0 to 300000:next
 440   if sc>hsc(lv) then hsc(lv)=sc
 450   cls:sp_set(0,0,0,0):wipe():home(0,0,0)
 460 endwhile
 470 /***********
 480 func bgscr()
 490   bg_scroll(0,bx1/2,0):bg_scroll(1,bx2/3,0)
 500   bx1=(bx1+1) mod 1024:bx2=(bx2+1) mod 1536
 510 endfunc
 520 /************
 530 func stage_set()
 540   int sx,sy
 550   for i=0 to 15:for j=0 to 14
 560     fill(256+j*ma,i*ma,256+j*ma+ma,i*ma+ma,gr1(j+i*16))
 570   next:next
 580   ex=15*ma-1:ey=16*ma-1
 590   get(256,0,256+ex,ey,gr2)
 600   fill(0,0,1023,1023,0)
 610   locate 0,0:print "STAGE";ro;"  LEVEL";lv;"  HI-SC";hsc(l
v)
 620   for i=0 to 3
 630     symbol(85-i,110,"STAGE "+str$(ro),1,1,2,7+i,0)
 640   next
 650   for i=0 to x68
 660     sx=256+rnd()*(768-ex):sy=rnd()*(1024-ey)
 670     put(sx,sy,sx+ex,sy+ey,gr2)
 680   next
 690   for i=0 to 2
 700     get(768,i*256,1023,i*256+255,gr2)
 710     put(0,i*256+256,255,i*256+511,gr2)
 720   next
 730   fill(768,768,1023,1023,0):fill(50,110,255,130,0)
 740 endfunc
 750 /***********
 760 func sprite()
 770   sp_init():sp_clr()
 780   sp_disp(1):sp_on(0)
 790   bg_fill(0,20):bg_fill(1,20)
 800   bg_set(0,0,1):bg_set(1,1,1)
 810   for i=1 to 14
 820     d=rnd()*64:gr1(d)=i:sp_def(i,gr1,0):gr1(d)=0
 830     sp_color(i,rgb(rnd()*25,rnd()*25,rnd()*25))
 840   next
 850   gr1(0)=15:sp_def(16,gr1,0)
 860   for i=0 to 1:for j=0 to 150
 870     bg_put(i,rnd()*64,rnd()*32,257+(j mod 14))
 880   next:next
 890 endfunc
 900 /***********
 910 func enemy()
 920   dim str e(7)[32]={
 930   "12222230122222301211123412222230",
 940   "12222234133333301222223412655230",
 950   "12222234111111301202023412222230",
 960   "13030334122222301202023412222230",
 970   "11010134122222301202023412222230",
 980   "12020234122222301262623412111230",
 990   "12222230122222300444440004444400",
1000   "23333345233333340233333452333340"}
1010   dim int pal(9)={63420,50736,38052,25368,20416,
1020                   63488,10272,20842,31412,41982}
1030   for i=0 to 7:for j=0 to 31
1040     gr1(j+i*32)=val(mid$(e(i),j+1,1))
1050   next:next
1060   for i=0 to 9:palet(i+1,pal(i)):next
1070 endfunc
1080 /*************
1090 func sound()
1100   f=fopen("SCRM_ML.PCM","r")
1110     fread(pcm1,19168,f)
1120   fclose(f)
1130   f=fopen("APL.PCM","r")
1140     fread(pcm2,62111,f)
1150   fclose(f)
1160 endfunc
1170 /***********
1180 func title()
1190   int c=0
1200   dim str w(1)={"ON","OFF"}
1210   for i=0 to 3
1220     symbol(10-i,60,"DOT PRO-68K",1,1,2,7+i,0)
1230     symbol(63-i,220,"PUSH START BUTTON",1,1,1,7+i,0)
1240     symbol(176-i,244,"By T.Morikawa",1,1,0,7+i,0)
1250     symbol(100-i,160,"> LEVEL "+str$(lv),1,1,0,7+i,0)
1260     symbol(100-i,180,"  WAIT "+w(wa),1,1,0,7+i,0)
1270   next
1280   repeat
1290     st=stick(1):tr=strig(1)
1300     if tr=2 then end
1310     switch (c+(st<>0))
1320       case 1:lv=lv+(lv<9 and st=6)-(lv>1 and st=4)
1330             for i=0 to 10000:next
1340             fill(144,160,160,170,0)
1350             for i=0 to 3
1360               symbol(148-i,160,str$(lv),1,1,0,7+i,0)
1370             next
1380             break
1390       case 21:if (st=4 or st=6) then wa=(st=6)
1400               fill(139,180,170,190,0)
1410               for i=0 to 3
1420                 symbol(142-i,180,w(wa),1,1,0,7+i,0)
1430               next
1440     endswitch
1450     if (st=8 or st=2) then { c=(st=2)*20
1460       fill(97,160,106,190,0)
1470       for i=0 to 3
1480         symbol(100-i,160+c,">",1,1,0,7+i,0)
1490       next
1500     }
1510     dotwait():bgscr()
1520   until tr=1
1530   x68=lv*100
1540 endfunc
```

仮想ドライバの開発実験PART7.

仮想ドライバの改良その3

電机本舗 由井 清人 Yui Kiyoto

今月は通信制御のために必要なこと，やらなければならないことを解説します。特に，SCCに関する情報を詳しく紹介していますので，SCC制御で悩んでいる人は参考にしてください。

今回は，主にRS-232C制御の高速化を行います。RS-232CはX68000ではZ8530SCC（以後SCCと表記）という通信用LSIで制御しています。

前回のおさらいになりますが，表1にSCCの内部構成を示します。SCCの制御そのものは，同じく表2の書き込みレジスタ，読み込みレジスタにCPUから命令を発行します。

SCCは，内部にA，B2組の通信制御機能をもっています。A，Bチャンネルは，若干仕様が異なりますがほとんど同じ構造です。ですから同じものが2つ入っているくらいに考えてよいでしょう。

通常使用できるのはAチャンネルのみです。Bチャンネルは，X68000ではマウス専用に割り振られていますので，通信制御には使えません。

SCCの簡単な動作図を図1に示します。

SCCの基本動作

SCCは1バイトのデータをパルスに合わせて1ビットずつRS-232Cポートへ出力していきます。これより，通信においてはパルスの発生が重要であることがわかると思います。通信のボーレートとは正にパルスの速さにより決定されるわけです。SCCはパルスの速度（厳密には周期）を可変にすることにより，300，1200bpsなどの各種の通信速度を作り出しています。これは計算式により導き出せます。通常，1バイトのデータはスタートビットとストップビットで挟んで送りますから，合計10ビットとなります。ですから，

ボーレート＝（1秒間における）転送ビット数

転送ビット数＝パルス数

表1 内蔵の通信チャンネルの機能

＊非同期モード
- 5，6，7，8ビット/キャラクタ
- 1，1.5，2ストップビット/キャラクタ
- 偶数パリティ，奇数パリティ，パリティなし
- x1，x16，x32，x64クロックモード
- ブレイクの生成と検出
- パリティ，オーバーラン，フレーミングの各エラーの検出

＊同期モード
- バイト指向同期モード
  - キャラクタ同期は，内部，外部のいずれも可能
  - 1または2個の同期キャラクタ
  - 同期キャラクタは6または8ビット
  - 同期キャラクタの自動挿入または削除
  - CRCの生成と照合
- SDLC/HDLCモード
  - アボートシーケンスの生成と検出
  - 自動ゼロ挿入と削除
  - メッセージ間での自動フラグ挿入
  - アドレスフィールドの検出
  - 情報フィールドの端数処理
  - CRCの生成と照合
  - SDLCループモードのEOP検出によるエントリ（オンループ）と脱出
- データ転送速度
  - 最大1.5Mbits/sec（モノシンク，バイシンク）
  - 最大375Kbits/sec（FM符号化方式DPLL）
  - 最大187Kbits/sec（NZRI符号化方式DPLL）

表2 制御レジスタ

・読み込みレジスタ

レジスタ	機能
RR0	送信/受信バッファステータスおよび外部ステータス
RR1	特別受信条件ステータス，端数コード，エラー条件
RR2	修飾割り込みベクタ（チャンネルBのみ） 非修飾割り込みベクタ（チャンネルAのみ）
RR3	割り込み保留ビット（チャンネルAのみ）
RR8	受信バッファ
RR10	雑ステータス，ループ/クロック情報
RR12	ボーレートジェネレータの時定数　下位バイト
RR13	ボーレートジェネレータの時定数　上位バイト
RR15	外部/ステータス割り込み情報

＊両チャンネル共通のRR2レジスタは，ほかの7本のレジスタと区別すること

・書き込みレジスタ

レジスタ	機能
WR0	各種モードのCRCの初期化および初期設定コマンド。レジスタポインタ
WR1	送信/受信割り込みおよびデータ転送モードの定義
WR2	割り込みベクタ（各チャンネルからアクセス）
WR3	受信パラメータおよび制御
WR4	送信/受信雑パラメータおよびモード
WR5	送信パラメータおよび制御
WR6	同期キャラクタまたはSDLCアドレスフィールド
WR7	同期キャラクタまたはSDLCフラグ
WR8	送信バッファ
WR9	マスタ割り込み制御およびリセット（各チャンネルからアクセス）
WR10	雑トランスミッタ/レシーバ制御ビット
WR11	クロックモード制御
WR12	ボーレートジェネレータの時定数　下位バイト
WR13	ボーレートジェネレータの時定数　上位バイト
WR14	各種の制御ビット
WR15	外部/ステータス割り込み制御

＊両チャンネル共通のWR2，WR3レジスタは，ほかの14本のレジスタと区別すること

という関係が成り立ちます。つまり，ボーレートはそのまま1秒間のパルス回数にほかなりません。たとえば，9600bpsであれば9600Hz（もしくは9.6kHz）のパルスを与えてやればよいわけです。パルスは通常はクロックジェネレータにより作り出します。

SCCの基本構造を簡単に表したものを図2に示します。ここにあるクロックジェネレータで，パルスを発生させ，ボーレートジェネレータを通してシリアルコンバータに供給しています。

通常，シリアル通信を行うときには，ボーレートに合わせてクロックジェネレータでパルスを発振させません。前もって，必要とするパルスの数十倍以上のパルスを発振させておき，これを必要に応じて分周させて必要なパルスを得ます。パルスジェネレータはたいていは水晶発振素子を利用し精度の高いパルスを生成しています。水晶発振素子は，水晶に電圧をかけると電気的に振動を起こす性質を利用した素子です。

振動回数そのものは，水晶に特殊なカットを施し固有の共鳴振動数（通常は周波数で表す）を与えています。ですから，この素子は周波数を可変に発振しません。ま

図1　SCCの動作図

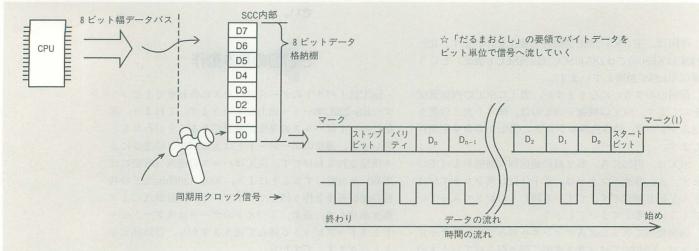



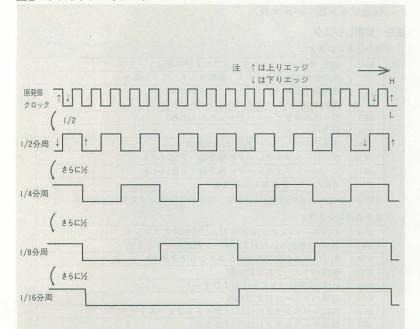
図3　クロックジェネレータ

・分周は，原発振クロックを数えるだけなので，精度は原発振クロックに依存する。
分周は通常1/2，さらに1/2というふうに行い，結果1/2，1/4，1/8，1/16のパルスを作る。これは，回路を作るとき，パルスの上りエッジか下りエッジかを調べるだけでよいので簡単になり都合がよい。なぜかというと電圧の上昇（もしくは下降）変化を検出するだけでよいからである

図2　SCCの基本構造

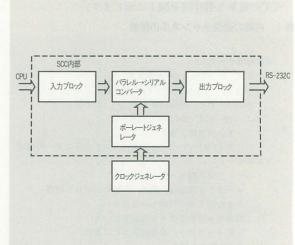
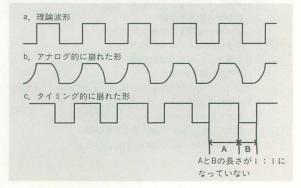



図4　クロックジェネレータの発振パルス

た，逆に可変であるならば，当然固定ではないので精度上の問題が出てきます。

このようなわけで，SCCに限らず，パソコンの通信処理は，クロックジェネレータで高い原発振を行っておき，これを分周して利用する方式をとります（図3）。

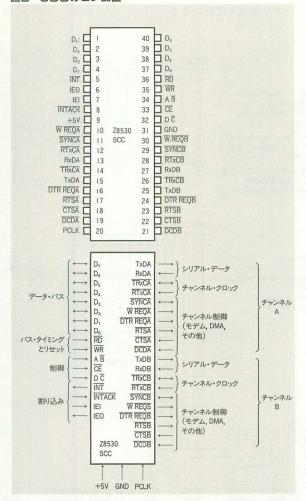
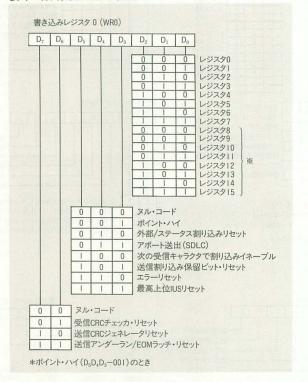
原発振周波数は，さらに，目的のボーレートの最低16倍以上に設計しておき，最低でも1/16に分周して使うようにしています。これは，水晶発振素子の発振周波数固定の性質とは別に，電子回路的な問題からの配慮です。図4にクロックジェネレータの発振するパルスを示します。aは，理論波形です。実際にはトランジスタの動作遅延，電子回路自身の持つキャパシタ（静電容量）より必ず波形は鈍り歪み，bのような形に崩れます。また，形そのものは歪まなくとも，cのようにON/OFFの比率（これをデューティサイクルと呼ぶ）が崩れ，信号としては，大きく歪む場合もあります。

このような理由により通信には，16倍ないし64倍のクロックを供給するようになっています。SCCにおいては，1，16，32，64倍のいずれかから選択できるようになっています。

図5 SCCのピン配置

表3 SCCレジスタアドレスマップ（全レジスタ，READ/WRITE可）

Ch No.	レジスタアドレス	D7 D6 D5 D4 D3 D2 D1 D0	備考
B	E98001H E98003H	←→ ←→	コマンドポート データポート
A	E98005H E98007H	←→ ←→	コマンドポート データポート

表4 WR0のレジスタ設定

コラム1 等価説明，もしくは嘘について

今回の記事の中で筆者は「等価説明」というあいまいな記述を行っています。これについて，蛇足ながら少し説明をします。

気づいている読者もいると思いますが，ときどき嘘の記述が文中にあります。厳密には嘘というより，文章を明快にするために，事実を四捨五入して記述しているというべきかもしれません。

今回の原稿では「2度めのときにはWR系とRR系の両方にマッピング……」という表現をしているところがあります。このあたりの機能はハードウェア的にどういうふうに実現されているかは調べていません。また，知る必要のないことでもあります。実現方法はいろいろ考えられます。本稿での記述もそのひとつでしょう。

実は，筆者は，SCCはコマンドポートへ初回の書き込みが行われたときに，SCCはWR0に指定されたレジスタ番号をRR系として認識し，コマンドポートアドレスにRRの内容を無条件に書き込んでいるのではないかと予測しています。

そして，2度めのアクセスが読み込みであればそれで終了。もし，書き込みであれば，コマンドポートに設定された値を該当のWRに設定して終わりです。このように考えれば，簡単なハード構成で同じ機能を実現できるはずです。

ただ，先に述べたように，実際の実現手法はソフトウェア技術者の観点ではどうでもよいことです。また，いま紹介した説明は，少しトリッキーで回路技術としては面白いですが，本来のSCCの動作説明からテーマが逸脱する傾向があります。

このような背景が書き手には存在すると思ってください。

図5にSCCのピン配置を示しておきます。イメージをつかむ参考にしてください。

SCC制御の実際

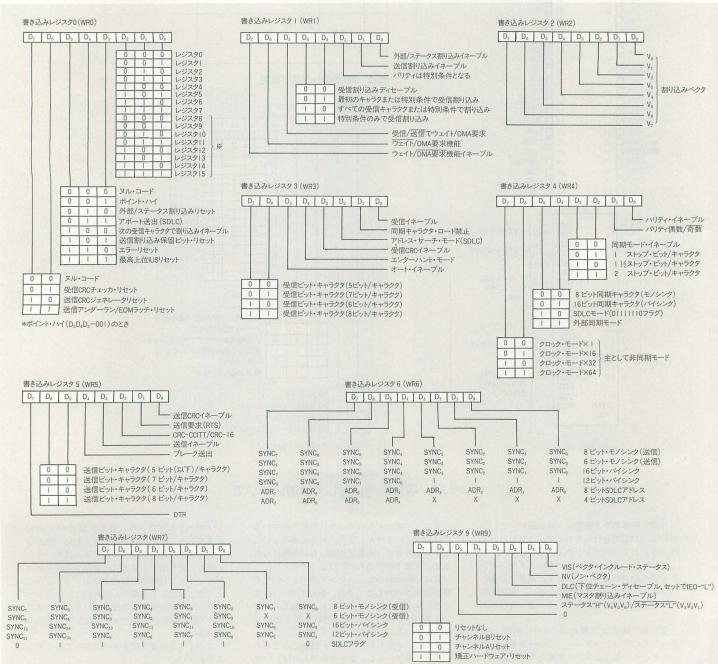
このようにSCCは，メカニックな存在ですが，実際には比較的簡単にプログラム制御できます。SCCはX68000上では，CPUのメモリ空間上の0xe98001から0xe98007のアドレスにマッピングされています。このメモリを読み書きすることにより自由にSCCを制御できるようになっています。表3に対応表を示します。

0xe98001，0xe98003はBチャンネル，すなわちX68000においてマウス専用として使用されているので，我々に関係するのは，Aチャンネル用の0xe98005と0xe98007のアドレスとなります。

コマンドポート（0xe98005）はSCCの制御用に使用されます。データポート（0xe98007）は，RS-232Cの送受信データの受け渡しに使用します。送信するときにはここに送信データを書き込み，SCCへ渡します。受信のときには，ここに受信データが格納されますからリードすればよいわけです。

コマンドポートの制御

SCCの制御は，表2の書き込み/読み込みレジスタへデータを設定して行います。

表5　SCCの各レジスタの詳細

SCCのわかりにくい点に，1バイトのコマンドポート(0xe98005)だけを利用して，16本の書き込みレジスタ(WR0-WR15)と9本の読み取りレジスタ(RR0-RR15)を制御することがあげられます。

SCCへの設定は，2バイト命令です。コマンドポートに対して，2バイト連続して設定を書き込むことにより行います。

まず，1バイトめの書き込みは，無条件にWR0レジスタに書き込まれます。表4にWR0レジスタの設定うちわけを表します。

WR0レジスタのD0〜D2ビットを見てください。実は，ここで，続く2バイトめの書き込みレジスタを指定しているのです。ですから，SCCに対する設定は，次の段階を踏みます。

1) WR0に基本設定と，設定したいレジスタの番号を指定する

2) 指定したレジスタに設定を行う

この，一連の制御で理解しづらいのは，初めの書き込みでは，コマンドポート(0xe98005)はWR0として動作し，2度目の書き込みでは，別のレジスタになっているということです。

コマンドポートの制御/読み込み

コマンドポートの使い方がだいたいわかってきたと思います。では，次に読み取りレジスタ(RR0〜RR15)の

割り込みレジスタ10(WR10)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- 6ビット/8ビット同期
- ループ・モード
- アンダーランでアボート/フラグ
- マーク/フラグ・アイドル
- ポーリングでアクティブ(GAOP)

0	0	NRZ
0	1	NRZI
1	0	FM1(送移1)
1	1	FM0(送移0)

- CRCプリセット1(SDLC)/0

書き込みレジスタ11(WR11)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

0	0	TRxC 出力−水晶出力
0	1	TRxC 出力−送信クロック
1	0	TRxC 出力−ボーレート・ジェネレータ出力
1	1	TRxC 出力−DPLL出力

- TRxC 出力/入力

0	0	送信クロック−RTxC 端子
0	1	送信クロック−TRxC 端子
1	0	送信クロック−ボーレート・ジェネレータ出力
1	1	送信クロック−DPLL出力

0	0	受信クロック−RTxC 端子
0	1	受信クロック−TRxC 端子
1	0	受信クロック−ボーレート・ジェネレータ出力
1	1	受信クロック−DPLL出力

- TRxC 水晶/水晶なし

書き込みレジスタ12(WR12)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

TC_0 TC_1 TC_2 TC_3 TC_4 TC_5 TC_6 TC_7 時定数下位バイト

書き込みレジスタ13(WR13)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

TC_8 TC_9 TC_{10} TC_{11} TC_{12} TC_{13} TC_{14} TC_{15} 時定数上位バイト

書き込みレジスタ14(WR14)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- ボーレート・ジェネレータ・イネーブル
- ボーレート・ジェネレータ源和PCLK/ RTxC入力
- DTR/要求機能モード
- オート・エコー
- ローカル・ループバック

0	0	0	ヌル・コマンド
0	0	1	エンターサーチ・モード(DPLL起動)
0	1	0	クロック欠如りセット
0	1	1	DPLLディセーブル
1	0	0	DPLLクロック・ソース−ボーレート・ジェネレータ
1	0	1	DPLLクロック・ソース−RTxC
1	1	0	FMモード・セット
1	1	1	NRZIモード・セット

書き込みレジスタ15(WR15)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- 0
- ゼロ・カウントIE
- 0
- DCD IE
- シンク/ハントIE
- CTC IE
- 送信アンダーラン/EOM IE
- ブレーク/アボートIE

読み出しレジスタ1(RR1)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- 全キャラクタ送出
- 端数コード2
- 端数コード1
- 端数コード0
- パリティ・エラー
- 受信オーバラン・エラー
- CRC/フレーミング・エラー
- エンド・オブ・フレーム(SDLC)

読み出しレジスタ2(RR2)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

割り込みベクタ

*チャンネルBでは修飾される

読み出しレジスタ10(RR10)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- 0
- オン・ループ
- 0
- 0
- ループへ送出中
- 0
- 2クロック欠如
- 1クロック欠如

読み出しレジスタ13(RR13)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

TC_8 TC_9 TC_{10} TC_{11} TC_{12} TC_{13} TC_{14} TC_{15} 時定数上位バイト

読み出しレジスタ15(RR15)

D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

- 0
- ゼロ・カウントIE
- 0
- DCD IE
- シンク/ハントIE
- CTS IE
- 送信アンダーラン/EOM IE
- ブレーク/アボートIE

使用方法を説明します。

さきほどは，コマンドポートに2バイト連続して書き込めば書き込みレジスタに設定できると説明しました。さらに，初回の書き込みで，2バイトめを設定する書き込みレジスタを指定すると記述しましたが，厳密には，次にアクセスするレジスタを指定するというのが事実です。レジスタを指定し，コマンドポートに書き込みが行われれば，WR0に指定したレジスタ番号は，WR系として扱われます。逆に，読み取りが行われれば，RR系のレジスタを指定したものとして働き，各RRレジスタの内容を取得することができます。これを，シンボル的に表現すると図5のようになります。コマンドポートは1度めのときには，WR0として働き，2度めのときにはWR系とRR系の両方にマッピングされるわけです。そして，メモリライトであればWR系，リードであればRR系レジスタとして動作するわけです。実際には，もう少し別の動きをしていそうですが，等価的にはこの説明で問題ないでしょう。

各レジスタの詳細説明

表5にSCCの各レジスタの詳細を示します。本稿の本来の趣旨に従うならば，ハードウェアの単なる仕様を多くの誌面をさいて掲載すべきではないかもしれません。しかし，利用する機能だけを紹介しても応用がききませんし，やはり読者の皆さんにもっとSCCについての知識をもってもらいたいと思い，あえて掲載しました。

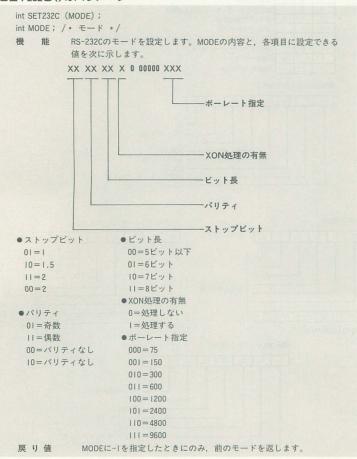



表6 SET232C()のパラメータ

int SET232C (MODE);
int MODE; /* モード */

機 能　RS-232Cのモードを設定します。MODEの内容と，各項目に設定できる値を次に示します。

XX XX XX X 0 00000 XXX

●ストップビット
01=1
10=1.5
11=2
00=2

●パリティ
01=奇数
11=偶数
00=パリティなし
10=パリティなし

●ビット長
00=5ビット以下
01=6ビット
10=7ビット
11=8ビット

●XON処理の有無
0=処理しない
1=処理する

●ボーレート指定
000=75
001=150
010=300
011=600
100=1200
101=2400
110=4800
111=9600

戻り値　MODEに-1を指定したときにのみ，前のモードを返します。

SCCプログラミングの実際

これまで，延々とSCCの動作や仕様を説明してきました。ここでは実際にプログラミングをしてみましょう。読者の方はここまでで，資料の多さに困惑されたと思います。ですが，実際のプログラミングは簡単なものです。多くの機能があったとしてもそのすべてを使用するわけではありません。該当する必要な機能だけを簡潔にコーディングすれば，驚くほどわずかなプログラムですみます。もちろん，該当する機能をきちんと把握すること，これが精神エネルギーをいちばん消耗するのはいうまでもありません。

SCCの初期化

まず，SCCの初期化からプログラミングしていきます。ここでは，Human68kに管理されているSCCをHuman68kから切り離さなければなりません。そして，切り離したあとは，SCCを自分のプログラム上から設定し直せばよいわけです。

リスト1がそのプログラム例です。関数_rs_inz( )の中で各種の初期設定を行います。

この関数は，パラメータとして通信速度を指定します。ここに，0（1200bps）から，6（76800bps）までの値をセットしSCCを呼び出します。_rs_inz()は諸々の設定とともに指定速度にSCCを初期化します。

1) RS-232Cの現在の設定の保存と初期化

リスト1 _rs_inz()

```
001:#define     BPS1200         0
002:#define     BPS2400         1
003:#define     BPS4800         2
004:#define     BPS9600         3
005:#define     BPS19200        4
006:#define     BPS38400        5
007:#define     BPS76800        6
008:
009:int __mode;
010:int _spd_tbl[] = { 128, 63, 31, 14, 6, 2, 0 };
011:/*********************************************
012:*   _rs_inz          初期化                 *
013:*   return  :        sts                     *
014:*********************************************/
015:_rs_inz( spd )
016:int spd;             /* speed          */
017:{
018:    __mode = SET232C( -1 );
019:    SET232C( 0x4c07 );
020:
021:    B_BPOKE(0xe98005, 0x03);    /* 受信モードセット */
022:    B_BPOKE(0xe98005, 0xe1);
023:
024:
025:    B_BPOKE(0xe98005, 0x05);     /* 送信モードセット */
026:    B_BPOKE(0xe98005, 0xea);
027:
028:
029:    B_BPOKE(0xe98005, 0x0c);     /* ボーレートセット */
030:    B_BPOKE(0xe98005, _spd_tbl[spd] );
031:
032:
033:    B_BPOKE(0xe98005, 0x0d);
034:    B_BPOKE(0xe98005, 0x00);
035:
036:    B_BPOKE(0xe98005, 0x01); /* ポーリングモードセット */
037:    B_BPOKE(0xe98005, 0x01);
038:
039:    int_a_sts = B_BPEEK( 0xe88007 );
040:    int_b_sts = B_BPEEK( 0xe88009 );
041:
042:    B_BPOKE( 0xe88007, 0x1e );
043:    B_BPOKE( 0xe88009, 0x00 );
044:
045:    return( 0 );
046:}
```

18行目を見てください。XCの標準ライブラリ関数SET232C()へパラメータを－1に指定してコールしています。SET232C()はRS-232Cの設定を変更する関数です。ただし，パラメータに－1を指定すると，機能が変わり，設定ではなく現在のRS-232Cの設定を返します。ですからここでは，現在の設定値をグローバル変数_modeに格納しています。この処理は，あとで仮想ドライブシステムを終了してSCCをHuman68kに戻すときに必要となります。Human68kに戻すときに当然SCCを最初の状態にしなければまりません。このときのために，最初の設定を保存しておくわけです。

19行目は本来なくてもよい設定です。SET232C()にパラメータ0x4c07を与えてRS-232Cを初期化しています。0x4c07は9600bps，ストップビット1，XON制御なしデータ長8ビットを表す値です。表6にSET232C()のパラメータを示しておきます。

実は，このあとにSCCに対して直に設定を行い，SET232C()で行った指定を翻していきます。なぜ，このような設定を行うかというと基本的な設定をXCのライブラリで行っておいて，要所要所を変更すれば，仮に設定し忘れのミスがあっても先のSET232C()での設定がSCCにかかっているので安全装置の役目を果たすからです。フェイルセーフへの配慮と思ってください。

2) 受信モードの設定

21～22行目で，SCCの受信モードの設定を行っています。21行目で，コマンドポート(0xe98005)にB_BPOKE( )を使ってバイトデータのメモリ書き込みを行っています。B_BPOKE( )は，やはりXCの標準ライブラリで，第1パラメータで指定したアドレスに第2パラメータの値をバイトデータとして書き込むものです。

21行目では，コマンドポートに0x03を書き込んでいます。ここで，表3を参照してください。コマンドポートへの最初の書き込みは自動的にWR0レジスタへの書き込みとなります。WR0は基本設定と次にアクセスするレジスタ番号の指定を行います。ここでは，基本設定にはヌルを指定しており，特に何も設定していません。ただし，0x03，つまりレジスタを指定する下位3ビットに3(2進法の011)をセットし，次のアクセスにWR3/RR3を指定しています。そして，22行目で，再びコマンドポートに書き込みを行っています。ですから，自動的にWR3レジスタに書き込みが行われたことになります。表5のWR3レジスタのところを参照してください。WR3には0xe3を書き込んでいます。表5により，8ビットキャラクタ（データ長）をセットしさらに受信機能をイネーブル

表7　各ボーレートに対するボーレートジェネレータの時定数

ボーレート	5MHzの場合の時定数	備　　考
76800	0(0000H)	
38400	2(0002H)	
19200	6(0006H)	
9600	14(000EH)	時定数＝$\frac{5\times10^6}{2\times(\text{ボーレート}\times16)}-2$
4800	31(001FH)	
2400	63(003FH)	
1200	128(0080H)	ただし，入力クロック(PCLK)を5MHzとし，データ速度の16倍のクロックを用いた場合とする。
600	258(0102H)	
300	519(0207H)	
150	1040(0410H)	
75	2081(0821H)	

コラム2　等価説明，もしくは嘘について　PartⅡ

ちょっと脱線しますが，再び四捨五入的説明について。コラム1に関するジレンマというのは，パソコン雑誌を見ていると必ず出てきます。筆者が，いつも疑問に思うものに8ビットCPUという記述があります。ちなみに代表的な8ビットCPUとしてZ80（別に6809でもなんでも同じですけど）の技術仕様を挙げてみます。はたしてこの石は8ビットCPUでしょうか。

<Z80>
・アドレス空間＝16ビット（0～0xFFFF）
・データバス＝8ビット
・各レジスタの幅＝8ビット（ペアレジスタを利用したときは16ビット）
・メモリ1番地の幅＝8ビット

基本的な仕様を見てみると，8ビットのところと16ビットのところが，混在しているのがわかります。特にアドレス空間は完全に16ビット仕様になっています。もしも，アドレスが8ビット仕様であれば，アドレス空間は0x00～0xFFすなわち256番地しか存在できませんから実用には供さないでしょう。ですから8ビットといえど，16ビットアドレスなのはよいバランスといえます。

結局，Z80には16ビットの側面もあり，完全な8ビットCPUとはいい切れません。厳密には8ビットのデータ処理と16ビットのアドレス空間をもったCPUということでしょう。しかし，実際には少しばかり違うことを承知で，8ビットCPUとパチリと簡潔に表現（定義）しています。

脱線しますが，よく雑誌，書籍での記事で「8ビットだから64Kバイトのメモリ空間しかない……」といい切ってしまうのは，間違いです。このような記述があるならば，その文章のライターはハードのことを理解していないと見るべきでしょう。

このほか，いくつかのCPUも含め仕様を見てみましょう。

<MN1613>
・アドレス空間＝16ビット（0～0xFFFF）
・データバス＝16ビット
・各レジスタの幅＝16ビット
・メモリ1番地の幅＝16ビット

Note:
松下製。どこから見ても完璧な16ビットCPUです。オフコンのCPUなどに以前採用されていました。
これはi8086と同時期の製品です。

<i8086>
・アドレス空間＝16ビット（0～0xFFFF）＋5ビットの複合指定
・データバス＝16ビット
・各レジスタの幅＝8/16ビット
・メモリ1番地の幅＝8ビット

Note:
No comment.

<MC68000>
・アドレス空間＝24ビット（0～0xFFFFFF）
・データバス＝16ビット
・各レジスタの幅＝32ビット
・メモリ1番地の幅＝8ビット

Note:
No comment.

このように見ていくと，i8086もMC68000もすべて厳密に16ビットといい切れないことがわかります。i8086は，8ビットと16ビット＋αの側面をもちます。MC68000にいたっては，8，16，32ビットの各側面をもっています。さらに，MC68000においても「メモリ1番地の幅」はいまだに8ビットです。この幅は，メモリの番地1つに振ったビット幅です。

いかがでしょうか。この事実を踏まえると，nビットCPUという具合にいい切るのは，かなりの心理的抵抗があります。ですが，実際には少々のことには目をつむり16ビットと定義しているしだいです。

このようなことは枚挙にいとまがなく，物事には必ず例外，特殊事情があり同様の事態を引き起こしていると思ってください。

(許可)にしています。第5ビットを示すオートイネーブルはCTS/DCDによるハードウェアフローを有効にするためのものです。

3) 送信モードの設定

25~26行目では送信モードの設定を行っています。さきの受信モードの設定と考え方は同じです。

25行目でコマンドポートでWR5を指定し,26行目でWR5に0xea(2進法11101010)を書き込んでいます。この意味はまず表5のWR5レジスタの説明を参照してください。具体的には,DTR制御有効,送信ビットキャラクタ8ビット(データ長),送信イネーブル(許可),RTS制御ONに設定します。

RTS制御をONにすると,送信データがないときには相手方に送信要求の信号を出すようになります。

4) 通信速度の設定

29~34行目も同様です。29行目での設定値0x0c(10進法で12)はWR12レジスタを指定しています。

また,WR12レジスタはWR13とペアになっています。ですから,33~34行目でWR13に対して書き込みを行っています。

WR12,13は通信速度を設定するレジスタです。ここには,SCCのボーレートジェネレータの分周比を決定する時定数を設定します。時定数とはクロックジェネレータから供給されたクロックパルスをカウントする回数のことと思ってください。クロックを時定数に従いカウントダウンし,必要とする通信タイミング(速度)を決定します。

図6にWR12,13とボーレートジェネレータの関係を示します。そして,時定数計算式を表7に示します。ここには,5MHzのクロック供給と分周比を16に設定したときの各速度の時定数をあわせて記載しておきます。

実際の時定数は,10行目にある配列変数spd_tbl[]にあらかじめセットしておきます。この値を,29~30行目でWR12へセットしています。WR13には,今回の場合常にゼロが設定値なので,33~34行目でゼロを書き込んでいます。

5) ポーリングモードへのセット

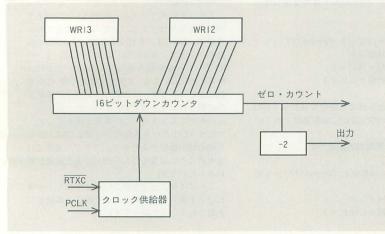
36~37行目まででWR1へ1を設定しています。詳しくは表5のWR1の説明を見てください。送信割り込み,受信割り込みすべてを禁止しています。ただ,例外的に外部/ステータス割り込みを許可します。外部/ステータスはDCD,CTS,SYNCなどの各端子の割り込み受けつけを行うものです。

39~43行目では,SCCではなくMFPと呼ばれる割り込み制御用のLSIの制御を行っています。39~40行目で,現在の設定を読み取り保存,そして,42~43行目で割り込み設定を変更してSCCとCPUの間に割り込みで連動しないように変更しています。MFPに関する解説は,今回は割愛し,次回に送ることにします。

図6 WR12,13とボーレートジェネレータの関係

SCCの使用終了,復旧

ポーリングによるSCC制御を行ったあとには,再びHuman68kへSCCを返却しなくてはなりません。

リスト2に,Human68kへの返却復旧の関数_x68_rcv()を示します。

7~8行目でWR1に,受信割り込みをイネーブル,外部/ステータス割り込みをイネーブルします。10~11行目はMFPの設定をあらかじめ保存しておいた変数int_a_sts,int_b_stsで上書きし,最初の状態に戻しています。

さらに,12行目で駄目押ししてHuman68kのRS-232C設定ルーチンSET232C( )で設定し直しています。変数__modeはあらかじめ,保存しておいたRS-232Cの初期設定値です。

1文字送出

リスト3にSCC直接駆動の1文字送出関数_rs_out()を示します。

この関数は,パラメータで指定したデータをSCCを直接駆動しRS-232Cへ出力します。

まず,15行目から22行目でループを構成しています。送信が成功するまで延々とループさせる構造をもっています。永久ループに陥らないように,100万回という野蛮な回数を設定しています。このあたりは,とくに根拠があるわけではなく,安易に設定しています。本来ならば,タイマーかなにかを設定しておき,3秒以上送信できないときにはタイムアウトするようにすべきでしょう。ただ,速度が命の特殊処理なので軽くいなしています。

ループ内部の最初の行である16行目で,コマンドポートにレジスタ番号0を指定しています。次行17行目でコマンドポートをB_BPEEK()関数でリードしています。B_BPEEK()は,XCの関数で指定したメモリ番地をリードするものです。これより,16行目でRR0レジスタを指定

リスト2 _x68_rcv()

```
001:/**************************************************
002:*    _x68_rcv()          通信割り込み設定の復旧      *
003:*    return  : char, or sts                        *
004:**************************************************/
005:_x68_rcv()
006:{
007:     B_BPOKE(0xe98005, 0x01);
008:     B_BPOKE(0xe98005, 0x11);
009:
010:     B_BPOKE( 0xe88007, int_a_sts );
011:     B_BPOKE( 0xe88009, int_b_sts );
012:     SET232C( __mode );
013:}
```

し，17行目でこれをリードした形になります。RR0に関しては表5を参照してください。

17行目は，リードした値を0x04で論理積をとり，D2ビットがONになっているかをチェックしています（0x04の2進値は，000000100)。もし，ONになっていればSCCの送信バッファは空を意味します。したがって送信可能と見なし，データポート（0xe98007）に送信データを書き込みます。これにより，SCCに送信データが転送されます。

また，SCCの送信バッファが空でなければ，前回の送信データはまだ相手に送られていないので，次のループにはいります。

1文字入力

同様にしてリスト4にSCC直接駆動の1文字入力関数_rs_in( )を示します。

この関数は，_rs_out()の入力版と思ってください。SCCより入力したデータを戻り値とします。

まず，12行目から18行目でループを構成しています。

ループ内部の最初の行である13行目で，コマンドポートにレジスタ番号0を指定しています。次行14行目でコマンドポートをB_BPEEK()でリードしていますから，RR0を指定，リードしたことになります。

RR0に関しては表5を参照してください。

14行目は，リードした値を0x01で論理積をとり，D0ビットがオンになっているかをチェックしています（0x01の2進値は，000000001)。もし，オンになっていればSCCの受信バッファには受信データが存在していることになります。したがって受取可能と見なし，データポート(0xe98007)より，メモリリードを行えば，SCCより受信データを受け取ることができます。これをやっているのが15行目です。

次回予告

初めSCCポーリング処理は，動けばいいだろうと考え，説明もなしにプログラムリストだけ掲載すればよいかと思っていました。結局，今回の記事を読んでもらえればわかるとおり，丹念にレポートすることになりました。

すべての資料を掲載しておくことができなかったのが心残りですが，必要十分な知識を提供できただけでもよしとしましょう。(「InsideX68000」も併読するとなお理解が深まるでしょう)。

結果論ですが，本来の仮想ドライブの開発記事を逸脱することになったとしても，周辺用LSIの制御方法をちょうどよく紹介できて自分なりに満足しています。これを機会に，パソコンのハードウェアに読者の方々が興味をもっていただけることを筆者は願っております。

さて，次回は今回説明できなかったMFPの扱いを解説していく予定です。そして，前回のDOSバージョン識別，そして今回のSCCの高速制御プログラムを仮想ドライブシステムに組み込み，総括したいと思っています。

参考資料

『マイコンピュータNo2インターフェースLSIの研究』CQ出版社

『Z80ファミリ・ハンドブック』額田忠之著，CQ出版社

リスト3 _rs_out()

```
001:/**/
002:/*******************************************************
003:*    _rs_out          R-S232Cへ 一文字送信           *
004:*    in      : c      送信文字                          *
005:*    return  : no                                        *
006:*******************************************************/
007:_rs_out( c )
008:unsigned char          c;
009:{
010:    long          i;
011:    int           sts;
012:
013:    sts = -1;
014:
015:    for( i=0 ; i<1000000L ; i++ ) {
016:            B_BPOKE(0xe98005, 0);
017:            if( B_BPEEK(0xe98005)&0x04 ) {
018:                    B_BPOKE(0xe98007, c);     /* 1文字送出 */
019:                    sts = 0;
020:                    break;
021:            }
022:    }
023:
024:    return( sts );
025:}
```

リスト4 _rs_in()

```
001:/*****************************************************
002:*    _rs_in       RS-232C一文字受信             *
003:*    return  : char, or sts                        *
004:*****************************************************/
005:_rs_in()
006:{
007:    long          i;
008:    int           sts;
009:
010:    sts = -1;
011:
012:    for( i=0 ; i<1000000L ; i++ ) {
013:            B_BPOKE(0xe98005, 0);
014:            if( B_BPEEK(0xe98005)&0x01 ) {
015:                    sts = B_BPEEK(0xe98007);
016:                    break;
017:            }
018:    }
019:
020:    return( sts );
021:}
```

コラム3 消費税導入で技術資料がなくなったわけ

表5に載せたSCCの資料は，4年ほど前にシャープの液晶事業部（当時のX68000関係のソフトハウス窓口，現在は名称が変わっています）の方からいただいたものの抜粋です。

当時，X68000における高速データ転送技術を確立するために，資料を探し回っていたのを覚えています。結局，SCC関連の書籍の一覧をシャープの担当の方から教えていただいたのですが，消費税導入にともなって市場からなくなっており結局入手できませんでした。もう，読者の皆さんはすっかり消費税に慣れて10年昔日のごとく感じておられると思います。

さて，なぜ消費税導入がSCCの書籍と関連があるのでしょうか。実は消費税導入後多くの書籍が市場から消えたのです。これは，書籍に消費税を算定した値段をつけねばならなくなったからです。よく売れている書籍であれば問題はないのですが，技術書のように特殊な（市場性のない）ものでは，消費税に対応した価格への変更ができず，結局のところ廃版もしくは処分という形で消えたと聞いています。このようなわけで，今回は少し詳しくSCC関連の資料を掲載しておきました（本当は全部掲載したいくらいです）。

また，シャープの担当者からいただいた資料は，SCCの技術資料の一部抜粋というものでした。オリジナルの出版物のタイトルなどは，恥ずかしながら不明です。このあたり了承ください。

PS．高木さん，お元気ですか！

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1995年1月18日の到着分までとします。当選者の発表は1995年3月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

1 スーパーストリートファイターII 2名

X68000用 5″2HD版 9,800円(税別)

カプコン

今月号からキャラ別紹介が始まったスパIIです。まだ持っていないキミ，これで遅れを取り戻すのだ！

2 違法コピー防止啓蒙キャンペーンポスター 10名

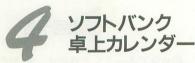
日本パーソナルコンピュータソフトウェア協会
☎03(3253)9166

パソコン違法コピーをしない・あげない・使わない，というキャッチフレーズで作られたキャンペーンポスター。千葉麗子ちゃんです。

3 MIYA-NET特製カレンダー 10名

MIYA-NET

MIYA-NETより今年も壁掛けカレンダーをいただきました。綺麗な色の動物イラストです。

MIYA-NETは，Z-MUSICをサポートしている草の根ネットです。無料のネットですが，得られるものがあったという方には，年間3,000円の寄付をお願いしています。
アクセスは☎048(650)1234 Tri-P CHXMIYA

4 ソフトバンク卓上カレンダー 20名

ソフトバンク

1995年のカレンダー。今年はCDサイズのケース入りです。

愛読者モニタ

モニタの応募方法

希望するモニタ記号をとじ込みのアンケートはがきの左下のスペースまたは官製はがきに記入してお申し込みください。応募の際に使用環境について明記する必要はありませんが，当選された方には，モニタとして使用ののちレポートを提出していただきます。締め切りは1994年1月18日の到着分までとし，当選者の発表は1994年3月号で行います。また，雑誌公正競争規約の定めにより，当選された方はこの号のほかの懸賞には当選できない場合がありますので，ご了承ください。

A 電話番号データベース「黒船」全国版 1名

CD-ROM版

62,000円(税別)

エンデバーズ
☎03(3409)3600

特集41ページで紹介したCD-ROMです。電話帳ハローページに掲載されている約5,000万件のデータを収録したものです。

11月号プレゼント当選者

1 F-Card V5 for x68k （石川県)川口 聡 （和歌山県)平野 純 **2** スターラスター （埼玉県)長崎 望 （静岡県)那須芳幸 （大阪府)竹内孝雄 **3** クイーン・オブ・デュエリスト外伝α＋ （岩手県)佐藤 学 （埼玉県)半田将義 （三重県)黒部浩孝 （岐阜県)大塚京吾 （福岡県)多和田真康 **4** CD「KALEIDOSCOPE」 （栃木県)大嶋靖浩 （愛知県)武藤信行 森本 真 **5** CD用キャリングケース「PAPARA」 （京都府)奥山貴士 （大阪府)溝渕剛 （岡山県)村上 晃 （敬称略）

以上の方々が当選しました。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。

石の言葉，言葉の夢

[第9回]

ダブルクリックのなくなる日

Ogikubo Kei
荻窪 圭

携帯電話を買った。ちょいと必要になったのと，それから安かったからだ。本当に安い。ちゃんと身分証明書を持っていれば，申し込んだその場で(30分くらいかかるけれども)電話番号ももらえる。加入料は３万円なにがしである。電話機は定価がだいたい７～８万円である。それが，全部込みで，電話機の定価以下の値段で買えるのだ。いまはもっと安くなっているらしい。もうこれは時価状態だ。誰もが携帯電話を持つ時代かもしれない。

だが，通話料は相変わらず，ダイヤルQ^{2}並みに高い。６秒10円だったかな。８秒10円だったかな。このへんはタイプによっていろいろあるのだけれども，これに加えて，月々の基本料金が4,000円とか7,000円とかかかる。これは高い。しかも，通話料はかけた人間が払うのである。当たり前といえば当たり前だが，他人の携帯電話をコールすると，その人がものすごい電話料金を取られるのだ。携帯電話を買う人間は「どうせ受けるのが中心でこっちからかけたりしないから，いいや」って思っている。事実，携帯電話からかけねばならないことは少ない。

いざ自分が持ってみると，いろんなことに気がつく。

たとえば，意外に使えない場所が多い，ってことだ。「携帯電話お断り」って喫茶店が増えてきたとかそういうわけではない。地下ではまずダメだ，ってことは承知している。地域である。カバーしている地域でも場所によって，通じやすいところと通じにくいところがある。オフィス街や繁華街ではちょっと奥まったところへ行こうが，建物の中だろうが，たいてい大丈夫だ。余裕で通じてくれる。小さな地域差が激しいのだ。聞くところによると，IDOよりDoCoMoのほうがそういうムラは少ないそうな。悔しいよね。DoCoMoが使える場所でIDOが使えないとなるとさ。

電波が届かないときと，電源スイッチをオフにしているときは同じメッセージが流れる。電源が入ってないか，電波が届かない場所にいます，って内容だ。どっちかわからないから困る。スイッチ切っていたでしょう，なんてあとでいわれてもなぁ。ちょうど地下鉄に乗っていただけなのに。

２番目には，意外に電池がもたない，ってこと。そりゃそうだよな。常に受信状態で電波を受けているんだからさ。たいていのモデルでは最も軽量のバッテリが付属してくるが，これがまた，もたないのだ。１日もたない。いや，半日もてばいいほうなのである。仕方がないから，Ｌサイズの大きくて重いバッテリーを追加で購入する。これなら１日くらいもつが，今度は重くてかさばる。困ったものである。ノートブックパソコンと同じやね。

３番目は，心臓によくない，ってこと。当たり前だがところかまわずかかってくるというのは心臓によくない。ドキッとする。電車の中だったりすると，周りの目を気にしたりして。しかも，携帯電話を携えて歩く人がどんどん増えているために，人混みで呼び出し音が鳴ったりすると，みんな「自分ではないか」って思う。誰の携帯電話に着信したのかわからんのだ。先日，恵比寿のガーデンプレイスへ遊びに行ってみたの

だが，視界の中に携帯電話所有者が6人はいた。カバンに入れている人も含めるともっといただろう。これ以上普及したらすごいことになりそうだ。常に電話のベルが鳴り続ける雑踏なのである。これもまた心臓によくない。電話なんてキライだぁ。

4番目は，電話中は周りの目が消え去るということ。他人が街中や電車の中で携帯電話使っていると，「あいつ，こんなところで電話なんかして，恥ずかしくないのか」と思っていたのだが，しゃあないのだ。まず，かかってきた電話を取らないわけにはいかない。取らないと鳴りっぱなしだし，うるさいからと鳴っている途中でスイッチを切ると，かけてきた人に失礼だ。いったん取ってしまうと，電話に集中しないと，周りの目なんて気にしているとちゃんと会話できない。周りはうるさいし通話音質もよくないから。集中すると周りは見えなくなる。人混みでもかまわずキスしてるカップルと同じなんだろうね。相手に集中すれば周りなんてどうでもいいの。

携帯電話にはいろいろと発見が多くて面白い。

PDAが本当に実用化される頃はどうなっているのかねぇ，って，それはそのうち考えよう。携帯電話はPDAに統合されていくと思ってはいるけどね。

ダブルクリック危うし

携帯電話の話はさておいて，パソコンの話。

ユーザーインタフェイスである。いま，パソコンのユーザーインタフェイスが変わろうとしている。大袈裟だけど。

パロアルト研究所が開発したウィンドウシステムにアップル社が改良を加え，デスクトップのメタファー導入やマウスのボタンを1つに減らしてポップアップメニューの代わりにダブルクリックを導入するなどして，現在のパソコン用ウィンドウシステムの基礎が築かれた。

それまではUNIXのGUIを見てもわかるとおり，右ボタンでポップアップメニューが当たり前だったのだ。でも，Macintoshはあえてそれをやらず(実際，初心者にWindowsを使わせると，マウスにボタンが2つある，ってことになかなか気づかないのだ)，ワンボタンマウスにし，プルダウンメニューとダブルクリックですべてをまかなおうとした。Windowsも原則としてそれを踏襲している。

だが，最近，「ワンクリック」で動作するボタンがどんどん増えているのだ。

ワンクリックで動作するボタンというと，これもまたアップルの「ハイパーカード」が思い起こされるが，あんな感じである。

まずは，いろいろなアプリケーションにおいて，プルダウンメニューの代わりに，よく使うコマンドをボタン化してパレットに並べた「ツールパレット」や「ツールバー」が流行する。これはワンクリックで作動するボタンだ。

この頃はまだよかった。次の段階は初心者の取り込みである。パソコンをはじめて使う初心者にとってみれば，パソコンの世界特有のルールは少なければ少ないほうがいい。そのひとつに「ダブルクリック」が挙げられたのだ。

そもそも，シングルクリックは「クリックした対象の選択」であり，ダブルクリックは「クリックした対象の実行」であった。いわば，「選択」して「実行」する手間をはぶくために，「選択」したあとにもっとも頻繁に行われるであろう「実行」をワンアクションで動作させるショートカットみたいなものだったのである。

で，実行する以外に選択肢がないのであれば，シングルクリックでそのまま実行してしまってもいいではないか？　ってことになる。

そういうわけかどういうわけか，各社が初心者向けと銘打って出してきたマルチメディア系パソコンのほとんどが，シングルクリックでコマンドを実行できるシェルプログラム(シェルが不謹慎であれば，簡易メニュープログラム)をつけてきているのだ。

ダブルクリック危うし，である。

そもそも，このシングルクリックでなん

illustration : Haruhisa Yamada

でも実行できる簡易メニューを標準搭載してきたのも，Macintoshなんだよね。パフォーマシリーズが搭載したAt EASEが最初なのだ。98CanBeなんか，MacintoshとAt EASEを意識したってのがアリアリなメニューを載せてきているほどだ。

確かに，ダブルクリックは便利な概念であるけれども，いきなりマウスを握った人には難しい。でも，1日使えばすぐ覚えられる類のものだ。それよりも，鬱陶しいのは，シングルクリックとダブルクリックのルールなき混在である。

98CanBeの場合，シングルクリックで起動する98ランチってメニュープログラムが付属してくる。アイコンをクリックすると，それに応じたアプリなり何なりが立ち上がる。たとえば，そこでコントロールパネルを起動してみよう。すると，ウィンドウの中に，アイコンが並ぶ。でも，コントロールパネルはWindowsのルールで動いているものだから，ひとつひとつのアイコンはダブルクリックしなければならないのである。見た目は98ランチのアイコンと変わらないのに，である。これは非常に鬱陶しい。

さらに，シングルクリックで実行するとはどういうことか，というと，「クリックミス」が許されないということである。間違えてダブルクリックすることは少ないが，間違えてシングルクリックしてしまうことは多い。ちょっとマウスポインタがずれたとか，1回余計にクリックしたせいで，2回目のクリックが別のアプリを起動してしまったとか，である。Windowsのような重い環境だと，それが非常に多い。鬱陶しいのである。

1本指の世界へ向かうユーザーインタフェイス

しかし，シングルクリックで動き出すアプリケーション，というのは，非常にわかりやすい。これなら，マウスを初めて持つ人でもとまどうことはない。確かにそうだ。ショートカット機能をボタンに割り当てて，1発でことを済ませるのは非常に便利である。

でも，この道はどこまで続くのだろうかと思うと，ちょっと気が滅入る。

ピアノでもキーボードタイプでもいいけど，最初は誰もが1本指で鍵盤を叩く。慣れてくると2本や3本になったり，1本指でもいろいろと技を駆使できるようになるが，ちゃんと5本の指を駆使するにはそのための訓練が必要だ。

パソコンのユーザーインタフェイスもそうで，とうとう1本指の世界へ降りてきてしまったのだ。マウスポインタが手だとして，いままでの各種メニューやダブルクリックを駆使して効率的にマウスを使っていたユーザーインタフェイスを3本指とすると，マウスを持ってボタンをただ押すだけ，というのは1本指でピアノを弾くのと同じである。

今後，こういった1本指インタフェイスは増えていく。必ずもっといろんなところへ波及していく。概念としては正しいと思う。世の中のあらゆる装置が「ボタン」式になりつつあるいま，パソコンの画面がそうなっても不思議はない。ダブルクリックなんてモニタ内仮想世界でのみ通じる概念なのだから。鬱陶しいのはルールなき混在の過渡期だけかもしれない。

あとは，MacOSやWindowsからダブルクリックがいつ消えるか，ってところへいくのだい。いつしか，ダブルクリックも伝説の技となってしまうのだ。わははは。普遍的な技はないのである。

もうひとつの流行「ドラッグ&ドロップ」

シングルクリックのボタン形式と並んで，もうひとつ流行しつつあるのが「ドラッグ&ドロップ」である。なんでもかんでもドラッグ&ドロップできるシーンにしてしまおうという動きだ。アイコンのドラッグ&ドロップから始まり，アプリ内で選択した文字列やグラフィックをドラッグ&ドロップできるようになり，アプリ内の別ドキュメント間でもできるようになり，別アプリ間でもできるようになりつつある。Macintoshもドラッグ&ドロップ機能拡張でもって，それに対応したアプリなら，なんでもかんでもドラッグ&ドロップできるようにならんとしている。たとえば，文書の一部をデスクトップ上にドラッグ&ドロップすると，その一部だけをファイルにして保存してしまう。逆に，デスクトップにあるファイルをアプリケーションのドキュメントにドラッグ&ドロップすると，そのファイルが挿入されるという技もある(そういえば，SX-WINDOWもそうだな)。

将来，ダブルクリックが廃れていくとしても，それはドラッグ&ドロップで補おうという流れになると思う。ドラッグ&ドロップは，感覚的ではあるけれども，慣れるまでは使いこなしにくい技だ。それがちょっとひっかかるが，みなさん，よいマウスを使いましょう，ってことで，とりあえずお茶を濁すことになるのかもしれない。それが具体的に，どんな世界になるのか。今度考えよう(こればっか)。最近，パソコンなんて全然知らないOLが初めて一体型パソコンを触る，っていう，とある雑誌の企画に参加したおかげで，機械慣れしていない人がどんなオペレーションをするか，ってのを目の当たりにしたからね。一筋縄ではいかないことがよくわかったのだ。

新製品紹介

X68000XVI用アクセラレータ

Xellent30

Kioi Makoto 紀尾井 誠

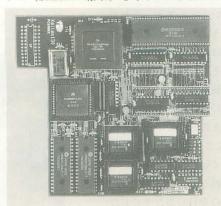
X68000XVIを32ビット化するアクセラレータです。33MHzのクロックでローカルRAM対応ですから，ピークスピードはX68030以上になる可能性もあります。

●

Xellent30はX68000XVI専用のアクセラレータです。要するに本体のCPUを引っこ抜いて代わりにアクセラレータボードをつけます。すると，ボード上の68030がCPUとして使えるようになります。基本的な考え方は本誌で石上氏の作成しているものとだいたい同じですが，68000モードとソフトウェアで切り換えができたり，256Kバイトのローカル RAMを装備していたりと，製品としての完成度はかなり高くなっているといえるでしょう。

ボード上に搭載されているCPUはX68030が使っているのと同じく680EC030(40MHz版)です。ただし68EC030用のソケットが採用されていますので，68030に載せ換えるということはできません。

本体のクロックを倍化しているので，16.6MHz時にCPUクロックは33MHz相当となります。68882コプロセッサには別系統で33MHzが供給されています。特に明言はされていませんが，クロックアップ時にはCPUのみ本体のCPUクロックの2倍で動作し，FPUは常に33MHzで動作することになります。

付属ソフトには起動時にどちらのCPUモードを使用するかという切り換えプログラム（SRAMに常駐）とそのメンテナンスプログラムがあります。それだけあれば，あとはHuman68k ver.3.0が必要な処理をすべて行ってくれます。

Xellent30（製品版とは少し異なります）

使用する場合は，まずHuman68k ver.3.0のシステムディスクは必ず用意します。それにはCACHE.Xが入っていますので，通常使用時には忘れずにCACHE ONに設定しておきます。また，せっかくFPUつきなのですからFLOAT4.Xも組み込んでおくとよいでしょう。

ツールなどのソフトではX68030で動作するのと同等のものがそのまま動きます。

アクセラレータ上で動作するゲームはHuman68k ver.3.0以降を使用したものに限られます（またはHuman68kを使用していないもの)。まあ，それ以前のものは従来のモードで動かせばよいのですから，まったく問題ないでしょう。

X68030対応のゲームなら68030モードでも起動できます。スーパーストリートファイターIIなどのCPUを区別しているものでは当然68030として認識されます。FLOAT4.XやREND030.X，XL/Image030専用版のような68030とコプロセッサを必要とするものも問題なく動作します。

ローカルRAMについて

SRAMといってもバッテリーバックアップされているわけではありません。あくまでも高速なRAMとして使われているだけです。

このSRAMはメモリエリアの最後部(VRAMの直前）にマッピングされており，特殊な方法でプログラムやデータを読み込むことになります。

ローカルRAMですからCPUとは32ビットのバス幅で接続され，こういったアクセラレータで致命的になるバスサイジングのロスがまったくなくなります。さらにSRAMですからノーウェイト動作が期待できます。事実，このSRAM上にプログラムを置いて実行すればアクセラレータが本来持っている速度(68030，33MHz)でプログラムが処理されます。これはX68030よりも速いモードです。

ただし，容量が256Kバイトとなっていますから，なんでもかんでもここに読み込んで実行するわけにはいきません。また，このエリアにはDMAの効力が及びませんので，実行するプログラムの選び方にも制限が出てきます。いずれにせよ特殊な方法でしかプログラムを読み込めませんので，現状では将来の拡張用と思っておいたほうがいいかもしれません。場合によってはIOCS ROMをここにコピーしてパッチ当てを行うことになります（なんかもったいない)。

本体に12Mバイトメモリフル実装したときにはアドレスが重なることになりますが，この場合は通常のRAMがマスクされます。

どの程度速くなるのか

ということで，核心にいきたいと思います。はっきりいって倍クロックで32ビットとはいっても，キャッシュが効いていないときには通常のX68000XVIよりもわずかに遅くなります。バス幅やメモリのウェイトが入ることなどを考えると，まあしかたないかなというところですが，石上版では「ちょっと速い」くらいにまでは持っていってるので，もうひとがんばりほしかったところではあります。

図1は，お馴染みstanfordベンチマークの結果です(XC ver.2.0，FLOAT2使用)。intで10MHz機の3.54倍，floatで2.45倍の速度になります。まあ，X68000XVIを10MHzで使っている人はいないでしょうから，16MHz比だとintで1.96倍，floatで1.38倍となります。

ざっと見て，普通の演算などをやらせたときはXVIの2倍程度の速度になります。画面表示などのI/O処理が入ってくるとまた違ったことになりますが，だいたいXVIとX68030の中間的な速度です。

X68000XVIの通常の使用状態でパワー不足が感じられる局面はそう多くないでしょう。パワーを使う事態，すなわち大規模なプログラムのコンパイルやTeXの作業，LHAの圧縮……こういった作業ではアクセラレータはフルにその実力を発揮します。

それ以外でパワーの足りないところとい

えば，やはりSX-WINDOW関係でしょう。さっそくSX-WINDOWを立ち上げてみましょう。見た目にはそれほど速くなった感じはありませんが……。これはVRAMへのアクセスはほとんど高速化されないことが原因でしょう。しかし，内部処理自体は軽くなっているはずです。

とりあえずSX-WINDOW上での画面書き換えの速度を評価するためSX-BASICでベンチマークテストプログラムを作成しました。これは一定回数のパターン表示を繰り返したときの時間を計るものです（リスト1）。なお，SUZUME.PT4は1994年3月号に掲載されたものを使用します。

結果は以下のとおり（単位は1/100秒）。

X68000XVI	16MHz	9774
Xellent30	33MHz	5885
X68030	25MHz	3871

VRAMのアクセスに一定の負荷がかかるので目に見えて速くなることはなくても，重ねあわせ処理などの負荷が高いところ（このテストのような）ではそれなりに威力を発揮しそうです。SX-WINDOWの場合，「速くなる」というより「重くならない」といった感じでしょうか。

そのほか，PCM8でフルPCMの曲を演奏していても通常のX68000XVIより速い状態を維持できます（データの程度にもよる）。

他機種への対応

今回の製品がX68000XVI専用というのは，主に基板形状の問題によるもので，10MHz機についても機種ごとに別基板のかたちで対応していく予定があるようです（ただし，CPU直付けのX68000Compactについては対応は不可能）。X68000は外見的にはほとんど変化のないデザインでしたが，内部配置は機種ごとにまるで違っていましたからどうしても共通化は難しいようです（仮に全機種で共通の基板にするとすると，H.A.R.Pくらいの大きさに収める必要があります）。さらに，初代機の場合は対応できないか，対応できたとしても，おそらく拡張スロットが1個使えないという状況になるはずです。

さらに，保証はされていませんが，CPUにはわざわざ40MHzのものが使用されていますのでだいたい本体のCPUクロックで20MHz程度までは動作するようです。24MHzになるとCPUより先に周辺回路がこけます。クロックアップ実験は今回行いませんでしたが，経験的にいって，XVIの場合はクロックに正比例してパフォーマンスが上がると思ってよいでしょう。クロックアップ改造はおすすめはしませんが，現時点で24MHzにしている人はクロックダウンの準備をしたほうがいいかもしれません。

問題点

CPUが68030化することと，高速化することでいくつかの問題点が挙がってきます。だいたいCPUのクロックアップと同じ条件ですので，それと同じような障害が発生することが予想されます。

よく知られているものに電源OFF時にFDのイジェクトができなくなるといったものがあります（イジェクトが完了する前に電源が落ちる）。これについてはIOCS ROMにパッチを当てて対処することは不可能ではありません。

ソフトウェアの互換性を見てみましょう。X68030で支障のあるソフトの大半はXellent30でも支障があります。すでに多くのツールがX68030に対応しているので，現在使用中の環境をそのまま移しても問題が現れることは少ないと思われます。編集室で使ってみた限りでは，Xellent30のみで問題になるソフトというのはありませんでした。

X68030導入への注意と同じものにはなりますが，念のため，考えられる問題点を挙げてみましょう。

68030CPUにはキャッシュがありますので自己書き換えするプログラムでは動作がおかしくなることがあります。代表的なところではLZXで圧縮されたツールなどが動作しないことがあります。

ラスターコピーがこけてスクロールすると画面にゴミが残ったりするのはCPUが高速すぎるのが原因です。適当な対処をしていないツールでは画面が壊れてしまいます（X68030で動けばほぼ大丈夫）。

話ではSxSIでフォーマットされたハードディスクをアクセスするときに問題があるということですが，すでに対応パッチプログラムも作成されています。

こういったもの以外の，プログラムの実行などの本質的な動作については特に問題がありません。日頃から，使用するシステムはできるだけ新しいバージョンのものを揃えるということに注意していれば問題は起こらないでしょう。

ハード的な問題点は，基板取り付けの安定性だけでしょう。アクセラレータの基板はCPUソケットでのみ固定されています。ですから，アクセラレータを装着したまま持ち運んだりしたときに大きな振動が加わると接触不良になる可能性もあります。する人はいないと思いますが，本体を横向きに置いたりするのもおすすめできません。

普通に設置して使っている分にははずれるようなことはありません。長時間の連続運転でも安定して動いています。

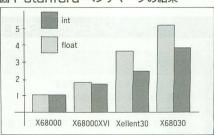
最後に

価格は59,800円（税別）ということで，まあまあのコストパフォーマンスだと評価できます。店頭価格ではもう少し安くなることも考えられますので，かなりおすすめのボードといってもいいでしょう。

そして，なによりうれしいのは，どうも最近ハードウェア関係の新製品は遅れがちになることが多いのですが，Xellent30については12月末あたりにはすんなり出てきそうだということです。

さて，ここまで読んで「10MHz機種に接続することは不可能なのか？」という疑問を抱いた人もいるかと思います。東京システムリサーチでX68000EXPERTのメイン基板を取り出し，無理やり装着したところ動作が確認されたようです。この場合は20MHz動作になります（本体の2倍）。しかし，ちゃんとした対応版が出てくるはずなので早まったことはしないでくださいね。

Xellent30　　59,800円（税別）
東京システムリサーチ　☎0425(28)1824

図1 stanford ベンチマークの結果

リスト1

```
▼Window Size (600,300),0,0,0,graphic test
int tim,w
di()
tim=A_line(&ha0af)
repeat
Bitmap1.move=w,56,0,0
w=w+3
until w>400
? A_line(&ha0af)-tim
ei()
end
 ▼1,Text1 (0,4,592,296),0,0,0,0,3,0,0,1,
func Text1_Click()
endfunc
 ▼8,Bitmap1 (0,56,356,184),0,0,0,0,suzume.pt4
func Bitmap1_Click()
endfunc
 ▼8,Bitmap2 (50,56,356,184),0,0,0,0,suzume.pt4
func Bitmap2_Click()
endfunc
 ▼8,Bitmap3 (100,56,356,184),0,0,0,0,suzume.pt4
func Bitmap3_Click()
endfunc
 ▼8,Bitmap4 (150,56,356,184),0,0,0,0,suzume.pt4
func Bitmap4_Click()
endfunc
 ▼8,Bitmap5 (200,56,356,184),0,0,0,0,suzume.pt4
func Bitmap5_Click()
endfunc
 ▼8,Bitmap6 (250,56,356,184),0,0,0,0,suzume.pt4
func Bitmap6_Click()
endfunc
 ▼8,Bitmap7 (300,56,356,184),0,0,0,0,suzume.pt4
func Bitmap7_Click()
endfunc
 ▼8,Bitmap8 (350,56,356,184),0,0,0,0,suzume.pt4
func Bitmap8_Click()
endfunc
```

ついに本編なし

以前からもがき続けてきた全機種共通システム。今回，ついに本編なしのTHE SENTINELのみとなってしまいました。

いままで，なんとか毎号掲載することを目指してきましたが，「シューティングゲーム作成法」の上杉氏が行方不明となり，原稿の依頼が不可能となって，ネタ切れとなってしまったのです。

Oh!X編集部でS-OSを使っている，またわかるスタッフもおらず，投稿作品もなし，現実問題として「ない袖は振れない」状態です。

毎月，THE SENTINELを楽しみにしていただいている方には申しわけありませんが，このような事態に至ってしまいました。

しかし，いい作品が投稿されれば，掲載するというこれまでのスタイルは変わりません。ただ，掲載できる作品がない場合には，このようにTHE SENTINELのみとなることをご了承ください。

もちろんS-OSへの投稿募集は継続しますし，いい投稿作品はどし掲載します。要するに，掲載するペースが落ちるだけです。もしも現在，投稿を目的に作品を制作されている人がいましたら，あきらめずに制作を続けてください。

ページは少なくなっても，このTHE SENTINELのコーナーが消滅することはありません，念のため。

MOOK化はどうなる？

本編がなくなってしまう。そうなると気になるのがS-OS "SWORD" MOOK化計画。まさか，本編のネタ切れをいいことに自然消滅を計る気では？　と考える人がいるかもしれませんが，そんなことはありません（もしもそんなことをしたら，編集部に爆弾を仕掛けられそう）。

せっかく，フリーソフト化に多数の読者の協力をいただき，マニュアルまで打ち込んでもらったのですから，皆さんの努力を無にするわけにはいきません。マニュアルのほうは順調に返送されてきているので，少し時間はかかると思いますが，順次まとめていきたいと思っています。

それから，締め切りを過ぎて協力スタッフに応募していただいた方もちゃんと，スタッフとして登録はされていますので安心してください。一応，第1段階の発注が終わっただけですので，また新たにマニュアル作成の必要ができた場合には優先的に連絡致します。

もしも，気長に待つのが嫌なのであれば，アンケートハガキにその旨ご記入のうえ返送して

S-OS "SWORD" MOOK化収録予定リスト

マシン語
- ×61部：デバッキングツールTRADE
- ×74部：ソースジェネレータSOURCERY
- ○77部：高速エディタアセンブラREDA
- ○93部：リロケータブルフォーマットの取り決め
- ○96部：リロケータブルアセンブラWZD
- ○90部：超多機能アセンブラOHM-Z80
- ○97部：リンカWLK
- ○99部：ライブラリアンWLB

インタプリタ言語
- ○28部：FuzzyBASIC
- ○85部：小型インタプリタ言語TTI
- ○92部：インタプリタ言語STACK

コンパイラ言語
- ○44部：FuzzyBASICコンパイラ
- ○60部：構造型コンパイラ言語SLANG
- ○81部：超小型コンパイラTTC
- ○89部：超小型コンパイラTTC++
- ○106部：実数型コンパイラREAL

エディタ
- ○69部：超小型エディタTED-750

ゲーム
- ○36部：アドベンチャーゲームMARMALADE
- ×40部：INVADER GAME
- ○41部：TANGERINE
- ○62部：シミュレーションウォーゲームWALRUS
- ○82部：TTC用パズルゲームTICBAN
- ○86部：TTI用パズルゲームPUSH BON!
- ○88部：SLANG用ゲームWORM KUN
- ○94部：STACK用ゲームSQUASH
- ○103部：ダイスゲームKISMET
- ○104部：アクションゲームMUD BALLN'
- ○113部：MORTAL
- ×115部：LINER
- ○116部：シミュレーションゲームPOLANYI
- ×117部：カードゲームKLONDIKE
- ○125部：SLENDER HUL
- ○129部：BLACK JACK
- ○133部：REVERSI
- ○135部：7並べ
- ○151部：B-GALETS2

開発ツール
- ○35部：MACINTO-C
- ×37部：テキアベ作成ツールCONTEX
- ○48部：漢字出力パッケージJACK WRITE
- ○78部：Z80浮動小数点演算パッケージSOROBAN
- ○79部：SLANG用実数演算ライブラリ
- ○110部：SLANG用NEWファイル入出力ライブラリ
- ○119部：COMMAND.OBJ
- ○123部：グラフィックライブラリGRAPH.LIB
- ○124部：O-EDIT&MODCNV
- ○145部：YGCS ver0.30

以上のリストはジャンルごとに分けて，「フリー許可　掲載部：アプリケーション名」の順に構成してあります。

ください。

フリーソフト化ありがとう

次に，新たにフリーソフトとして協力していただいたものを発表します。

まず，個性的でコンパクトな言語を発表してくれた平井真二氏です。

第81部　超小型コンパイラTTC
第85部　小型インタプリタ言語TTI
第89部　超小型コンパイラTTC++
第92部　インタプリタ言語STACK

どれも，ちょっとしたツール作りに威力を発揮する言語たちです。平井さん，ご協力感謝します。

次に，片岡正博氏です。いままで発表したゲーム3本の，フリーソフト化に協力していただきました。

第36部　アドベンチャーゲーム
　　MARMALADE
第41部　TANGERINE
第62部　シミューレーションウォーゲーム
　　WALRUS

片岡氏はOh!Xを卒業してしまったということですが，これらの投稿作品が掲載されたことが自信となり，とあるソフトハウスで元気に働いているそうです。ちょっと寂しい気もしますが，がんばっていいゲームを作ってくださいね。

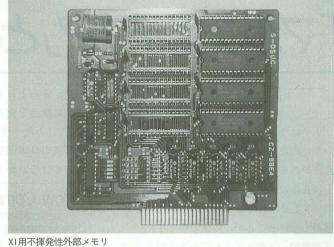
最後はS-OSユーザーズクラブで，現在もガシガシ活動を続けている黒木淳一氏。彼が発表した，

第123部　グラフィックライブラリ
　　GRAPH.LIB
第124部　O-EDIT&MODCNV

以上のアプリケーションをコピーフリーとしていただきました。

「現在，第7回CGAコンテストに向けてS-KALGō SYSTEMを開発中です。さらにGRAPH.LIB,O-EDIT,MODCNVが大きくパワーアップする予定です。あと今年中にとりあえず，新マニュアルを完成させるつもりですが，ダメですか？　あ，それからLoversの奥野さんが，GRAPH.LIBのポリゴン部だけをパワーアップしたPOLY.KITを制作しているのですがすごいですよ。あのスタークルーザーが動き（しかもクレイジーシェーディングで），そのうえとても速いです」

なんだか，すごそうな内容のハガキ。担当者としては，ぜひとも現物を見たいですね。それに「ダメですか？」なんて消極的な態度はいけません。自信をもって投稿してみてください。

そして，POLY.KITも気になるところ。黒木さん，GRAPH.LIBの新バージョンとともに期待して待っています。

X1用不揮発性外部メモリ

ちょっと宣伝

ここで，S-OSユーザーズクラブから怪しげな包みが届きました。なになに……中身は，X1用不揮発性外部メモリ？　一緒に送られてきたボードを見ると，なかなかしっかりした作り。ぜひ，Oh!Xで紹介して広くX1ユーザーに見てもらいたといいうことなので，ここに紹介します。

S-OSユーザーズクラブでは，オリジナル「X1用不揮発性外部メモリキット」の配布を行います。特徴は以下のとおり。

1）　純正EMM上位コンパチ
2）　X1の電源を切っても内容は消えない
3）　EMMからのIPLが可能
4）　容量1Mバイトまで増設可

配布価格は，基板のみが5,800円，基板+パーツ（RAM512Kバイト）が18,000円です。入手希望の方は，住所，氏名，と必要セットを明記のうえ，80円切手を同封して下記までご連絡ください。折り返し，案内状を送付します。ただし，希望者が20名に満たない場合は，配布できませんのでご了承ください。

〒536　大阪府大阪市城東区鴫野東1-13-18
森　喜一郎

現役でX1を使いこなしている人で，興味のある方は問い合わせてみては？

S-OSの今後

結構，つらい状況になってしまった全機種共通システム。はっきりいって，S-OSの活動自体，個人サークルのほうが，活発的な動きを見せています。

そこで，これからのTHE SENTINELでは，現在S-OS，Z80で活動を行っているサークルからの情報を積極的に掲載します。

会員募集でもかまいませんし，「こんなもの作ってみたんですけど，THE SENTINELで紹介してもらえませんか？」などありましたらどんどん，投稿してみてください。今回のS-OSユーザーズクラブのような，ハードを配布したい，というようなものも考慮します。

つまり，THE SENTINELをZ80ユーザーのための交流の場として開放します。Z80をまだ現役で使っている方は，ぜひ，開放されたTHE SENTINELを活用してください。投稿をお待ちしています。

1994■インデックス

第88回　知能機械概論――お茶目な計算機たち――

「マルチメディア」という言葉の奥に

「万能＝役立たず」

まずはマルチメディアという言葉です。いろいろな場面で使われます。いろいろな意味があるのでしょう。言葉の定義からいえば複数のメディア(媒体)ということで終わってしまうのでしょうが，それだけでは終わらないところがこの世の中です。

「複数の」メディアとわざわざ表現する場合に隠れている気持ちは「単数ではないんだエラいんだぞ」ということです。いうまでもなく，従来はひとつのメディアだったのだけれども，今度はそういう制限がなくなって複数のメディアが使えますよということですね。

しかし，マルチメディアという言葉にバラ色のイメージをもたせて使われている場面でちょっと気になってしまうのは，複数にすれば本当によいのかということです。むろん，原理的にはマルチはモノ/シングルを含んだ概念ですからマルチがいいのに決まっているという理屈も成り立ちますが，そう簡単には割り切れません。

そもそも，メディアというものは人間にとって総体としてはマルチであるのが自然であるという前提があります。人間の五感は統合化されて使われるのが本来の姿であるということです。だから，あらためて1つが2つや3つになったからといって，マルチというほどのことがあるのかということなのです。

そして，モノ/シングルにわざわざ限定している環境にも，それなりのきっちりとした意義が見出せる場合もあるのではないかと思ってしまうのです。単に無思想にメディアの本数をいくつか増やすのではむしろ逆効果となる場合があるのではないかと感じます。

基本的には，どんな場面でもなにかするときにメディアは多く，また複合化/統合化されていたほうが，情報伝達の効率から考えてもベターだというのは正しいでしょう。しかし，そのときの状況において，もっとも適したメディアというものがあります。そして，そのまわりを補助するメディア群が柔軟に構造を変えてくれるようなものでないと，「万能＝役立たず」という相変わらずの等式の具体例をまたひとつ増やしてしまうのではないかと思うのです。

How Computers Works

「How Computers Works」というMacintosh用のCD-ROMが，きわめてマイナーな雑誌のプレゼントに当たって送られてきました。このソフトは計算機がどういうふうに動くのかという原理をハードウェア面を中心に学習するものです。かなり期待をこめて実行したのですが，結果からいうと，それほど期待に沿うものではありませんでした。

計算機を構成する各要素について音声や文字によって順番に筋道だてて詳しく説明してくれます。それなりにためになるものなのでしょう。ただ，基本的には，動かない写真がポンと画面に出ていて，それに対する説明が音声で続くというパターンで，もう少し気がきいてもいいんじゃないのと感じます。

マンツーマンで人からなにかを習うとしましょう。その場合，写真1枚を出して長々と一本調子で説明しませんよね。相手の理解度に応じて話すスピードを変えたり，話をはしょったりしますね。

そういった学習者に対するアダプティブな(適応性のある)ところがこのシステムには欠けているのです。アダプティブになるには当然インタラクティブ(対話的)であることが必要条件です。学習者の状態をきめ細かにつかむには，学習レベルに関するデータを適宜収集しなければならないからです。

人が人に教える場合のコミュニケーションは，いちいちネーミングするまでもなくマルチメディアです。手取り足取り，雰囲気，声，図解，身振りなどで正確に相手の状態を察知したり教えたりすることができるからです。

コンピュータは，本のように文字だけでなく，画像や音声出力，あるいはマウスやキーボードによる入力が加わったのですから，それなりにマルチメディアと呼べなくもないでしょう。しかし，本来の人対人のマルチメディアに比べたら相変わらずメディア的に大きな制限が加わっているという事実にあまり変化はありません。メディアの数というよりはもっと本質的なアダプティブとかインタラクティブな問題であるともいえますが。

たまたまある雑誌にこのソフトが紹介されているのを目にしました。そのページに「マルチメディアの世界」と堂々と銘打たれているのを見ると，「やっぱりちょっと違うんじゃない？」という気持ちになってしまいます。あるいは，書いた人はどこを見渡してもバラ色に見える幸せな生き物なのでしょうか？

アメリカの陰謀説

研究開発の現場だけでなく，こいつはお金のなる木になりそうだと思う人々がいろいろなところで「マルチメディア」をキーワードとして，なんやかやと煽りたてています。本屋にもマルチメディア関連の本は山積みです。

ときどきパラパラとそのような本をめくってみるのですが，なぜか，いまいちその実態をつかんだという気になることができません。手放しで絶賛しているようなタイプの本(こういうタイプの本がほとんどですが)の場合は特にそうです。

ひとつだけとびっきり強烈な文章を発見しました。工業デザイナーの川崎和男氏のエッセイ(参考文献1)です。そのなかではマルチメディア本を血祭りに上げ，「マルチメディア」ブームの虚像を暴いています。

マルチメディア本では，企業の再活性化として，リエンジニアリング，リコンストラクチャリング，リストラクチャリングの3つをまず挙げるのだそうです。そして，川崎氏はこの3つの「リ」で始まる言葉を使った(やや苦しい)だじゃれで次のように揶揄します。

会社のマルチメディア事業化のなかで，

「リス」のようにおとなしくするか強気の「トラ」になって知ったかぶりをしなければ，会社に「リエン」と「リコン」を迫られるのだとマルチメディア本の著者は読者を脅迫しているのだと。

そして，儲けているのは，マルチメディアの本を書き，講演をし，企業でコンサルタントをしている連中だと書いています。また，マルチタレントっぽい人が，マルチメディアという言葉を通して本というメディアで儲けているのだとも書いています。

そして，川崎氏のエッセイはクライマックスに達します。「もっと赤裸々にいっておかねばならない。マルチメディアは米国の仕組んだマルチ商法だ」。

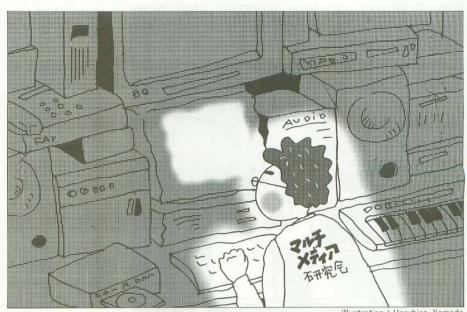
マルチメディアという言葉が指す時代の流れ自体は川崎氏も認めています。ただ，マルチメディアという言葉は使わずに「メディア・インテグレーション」という言葉に置き換えて使っているのだそうです。メディア統合のほうがより積極的な意味合いがあるのは確かなことですね。

illustration：Haruhisa Yamada

アングラ用語だった？

マルチメディア，マルチメディアと最近特に騒がしいのでこの言葉も最近作られたのかな(といっても10年ぐらい前からは耳にしてましたが)？と思っていましたが，それは間違いで，ほぼ僕が生まれたころの古い言葉なのです(参考文献２)。

1962年ごろから，サンフランシスコ州立大学などで行われていたサイケデリック・イベント，要するにスライドショーやライトショーをロックのライブ演奏と組み合わせたものをマルチメディアと呼ぶようになったのです。最も古い使用例はヒッピー文化誌「サンフランシスコ・オラクル」のサイケデリックイベントに関する記事だそうです。

1965年に刊行開始された「プレンティス・ホール音楽史シリーズ」のなかでソーズマンは次のようにマルチメディアを定義しています。「マルチメディアとは，多種多様な表現メディアを融合させ，さらにテクノロジーを駆使した，一種の環境芸術を指す」。

いまでは完全に技術用語のようになっているマルチメディアはもともとは前衛的で対抗文化(カウンターカルチャー)的な言葉だったというのですから驚きです。要するに，ポップアーティストのアンディ・ウォーホルとロックグループのヴェルベット・アンダーグラウンドが行ったEPIと呼ばれるショーこそ，マルチメディアだったのですね。海賊版ビデオをもっているので今度もう一度見てみましょう(熱中した時期があったのです)。

やっぱりアップルが

そのような言葉がなぜ技術用語となっていったか？　そこにアップルがからみます。アップル社は一世を風靡したMITのメディアラボに対抗して，マルチメディアラボなる研究所を1984年につくりました。そして，マイクロソフトがオペレーティングシステムの拡張規格をマルチメディアエクステンションと命名して，言葉が定着したとのことです。

ヒッピー上がりのサイケな若者がつくったアップル社が「マルチメディア」という言葉を使ったのが意図的だったのはたぶん間違いないでしょう。反体制，ドラッグ，LSD，反戦平和，ヒッピーという流れを少なくとも精神的には(初期の)アップル社が引き継いでいたことは事実ですから。

そういうわけで，マルチメディアという言葉には，カウンターカルチャー的な意味合いが濃厚に含まれていたわけで，現在でもアメリカではその言葉のそのような背景は意識され続けているというのです。まったく意外なことですね，これは。

インターネットとマルチメディア

インターネットという言葉が登場します。これはマルチメディアとは違い，現に存在する国際的なネットワークを表す固有名詞ですから，使う人によって定義がバラバラで混乱するようなことはありません。

そして，実際われわれの生活を大きく変革しようとする気配があります。アメリカではすでに「なんでもかんでもインターネットにつなごう」状態のようです。

たとえば，あのミック・ジャガーが前からネットワークに入れ込んでいるのは有名ですが，なんとローリングストーンズのコンサートをインターネットでライブ(なま)で聞けるのだそうです。

はたまた，ピザの宅配から始まり大学キ

「マルチメディア」という言葉の奥に

夏の猛烈に暑い日，高校生のための大学説明会(名古屋大学情報文化学部)が開かれ，Y先生と僕は「マルチメディア」パソコンシステムを実演してみせるということになりました。

それで，ビデオを使ってコンピュータグラフィックの作品を見せたり，見学に来た高校生をビデオで撮ってMacintoshに取り込み，Photoshopでいろいろ加工してから，カラープリンタで出力してプレゼントするなどをしましたが，もうひとつの出し物が,「マルチメディアとインターネット」に関連するものでした。

これはY先生の素晴らしい発案だったのですが,インターネットでアメリカのMTVにアクセスして最新のロックのデータを取り寄せて，その場でバーンとステレオで再生しようというものです。

2日あったうちの初日のほうは，何カ月も接続がおかしいという状態がズルズル続いていた影響を受けて不調に終わりましたが，2日目はなんとかロックミュージックをワークステーション室に響かせることができました。

50度にもなろうとする地獄のような部屋(もうすぐエアコンはつくでしょうね)に詰め込んで申し訳なかったのですが，われわれの努力が少しは伝わってくれたのではないかと信じている次第です，ハイ。

音楽業界の革命

高校生の前で見せたこのデモンストレーションなんぞは，ほんの子供だましといわれればそのとおりなのですが，音楽関連だけを見ても，いろいろと面白いことが起こっているようです。たとえば，インターネット・アンダーグラウンド・ミュージック・アーカイブ(IUMA)がそうです(参考文献4)。

IUMAは，マイナーなグループの曲のデータを蓄積しており，無料で聞くことができます。バンドは数千円の寄付を払おうと払うまいと原則として無料でさまざまなデータと曲を登録できるのです。

しかも，IUMAは，音楽業界から，レコード店とレコード会社を排除したいという明確な意思をもっています。クレジットカードの番号を送り，データを送ってもらうようにすれば，ビジネスとしても成立するわけです。ミュージシャンはダウンロード回数に応じたお金を受け取るのです。

このような音楽業界が成立すれば，リスナーは自分が本当に聞きたい音楽の一部をまず無料で聞くことができます。レコード会社の宣伝に依存しません。そして，これをもっと聞きたいと思ったら正式に曲をダウンロードし，クレジットカードを通じてお金を払えばよいのです。

その値段は大幅に安くなるでしょうし，支持されるミュージシャンに入るお金は逆に増えるでしょう。なぜなら，業界が間に入らなくなる分も浮きますし，メディア(CDなど)の材料費，流通経費などもいらなくなるからです。

う～む，こういう話を聞くと，マルチメディアってやっぱりバラ色の未来を作ってくれる夢の技術なんだな，と浮かれたくなってきますね，いかんいかん。

ャンパスの飲み物の自動販売機までが接続されているのだそうです。もちろん，自動販売機が自動で配達してくれるわけではありません。飲み物の在庫があるかとか，何度に冷えているかという情報が得られるのだそうです。

日本でも首相官邸にアクセスできるようになりましたし，NIFTY-Serveなどのパソコン通信との接続がなされ，メールだけではなく，インターネットのニュースもパソコン通信側から読み書きできるようになってきました。

大学説明会

マルチメディアはこのインターネットと融合して初めて将来が開かれるのではないかと思います(技術的な話は参考文献3などにまとめられています)。いろいろな面白い情報は世界各地に分散していて，それをネットワークで結びます。そして，それをどう見せるかというところにマルチメディアの真価が問われるのです。世界レベルのネットワークであるからこそマルチメディアの基準が確立されやすいという面も忘れてはなりません。

参考文献

1)川崎和男，“Design Talk II 19，マルチメディア・テオリア”，MacPower，1994年11月号，318-319pp.，アスキー
2)福富忠和，“愛と自由のパーソナル・コンピュータ”，STUDIO VOICE，1994年12月号，16-17pp.，インファス
3)丸山充，“インターネットにおけるマルチメディア技術”，bit，1994年11月号，4-14pp.，共立出版
4)“IUMAインタヴュー”，STUDIO VOICE，1994年12月号，38-39pp.，インファス

e-mailアドレス
ari@info.human.nagoya-u.ac.jp

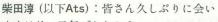
こちらシステムX探偵事務所

FILE-XVIII

社会科学系シミュレーションの系譜

Shibata Atsushi 柴田 淳

illustration : T.Takahashi

しばらくの間留守が続いた「システムX探偵事務所」に，いつものメンバーが戻ってきました。どうやら今回からは「シムシティー」のような社会科学系のシミュレーションに挑戦するようです。さてさて，どんなことになるでしょうか。

柴田淳（以下Ats）：皆さん久しぶりに会いますけど，元気でしたか？

琴張春香（以下春）：もちろん。

琴張護（以下護）：当然です。

マスター（以下M）：なんとかね。

Ats：そういえば，こんなに長い間どこに行ってたんですか？

春：ナ・イ・ショ。

Ats：まあどうでもいいんですけど……。ところで，昔「医は算術なり」というゲームがあったのを知ってますか？

春：「医は仁術なり」っていう言葉なら聞いたことあるけど。

Ats：これ，MZ-80B/2000シリーズ用のゲームで……。

M：誰も覚えているわけないじゃないですか，そんな古いこと。

護：どのようなゲームなのですか？

Ats：ええと，大まかにいっちゃうと開業医の経営シミュレーションなんです。

春：経営シミュレーションっていうことは，「ライフ&デス」みたいにメスを使って手術をするようなゲームではないわけね。

M：まあ，なんといっても昔のゲームですから。

Ats：最初に軍資金を1億円ほど渡されて，診療所を建設するところからゲームが始まります。で，それからその診療所をきりもりして，一定期間のうちにどれだけお金を貯められるかを競うんです。

M：とはいってもそこはゲームだから，プレイ中にはいろいろなイベントが起こるわけです。

春：たとえば宝くじに当たったり，親戚からの思わぬ遺産が転がり込んだり。

護：怪しい新薬を投与して患者に告訴されたり。

Ats：いや，さすがにそこまでどぎついイベントはありませんけどね。だけどこのゲーム妙にリアルで，それなりに利益を上げようと思ったら，患者に薬をいっぱい出さなければならない。つまり，良心的な診療をしていてはお金が貯まらないようになっているんです。

M：確かに，診療に対して支払われる人件費なんてタカが知れているでしょうからね。

護：この手のゲームなら簡単に作れてしまいそうですね。

春：そうよね，なにかイベントが起こったときも文章を表示するだけでいいでしょうし，診療所経営の収支計算もそれほど難しくなさそうだし。

Ats：確か，リストでは税金の計算なんて総収入に0.1を掛けるだけでしたよ。

M：そんな単純なものでも楽しんでいられる幸せな時代だったんですね。

Ats：でも，現実に行われている経理計算だって，それほど複雑ではないですよね。ただ考慮すべき項目とか例外のたぐいが多い，というだけじゃないですか。

護：なるほど。そうしてみると，経営シミュレーションというのはコンピュータゲームにしやすい題材かもしれません。

Ats：「医は算術なり」のあとにも，やはり経営シミュレーションゲームは頻繁に登場します。時代の移り変わりとともに，画面がハデになったり起こるイベントの数は増えたけど，基本的な部分ではそれほど変わってはいません。

第1世代のゲームたち

Ats：便宜上，初期の経営シミュレーションのようなゲームを「第1世代」と呼びましょう。この第1世代のゲームシステムというか，ゲームの元となっている手法を図式化してみることにしましょうか。

春：図にするっていうと？

Ats：それほど難しいことじゃないんですけどね。ええと，まず投資とか来客数などの入力があるわけですよね。

M：なるほど，そういうことですか。するとその入力に対して，天候条件や税率といったものを考慮した計算を行い，出力を得るわけですね。経営シミュレーションなら，出力はお金ということになります。

護：ただし，出力には予期せぬできごとなどによって多少の「ゆらぎ」があるかもしれません。

Ats：これを一般化すると，図1のようになると思います。投資額ばかりでなく，自然条件や景気といった複数の入力に，特定の計算モデルを適用する。さらにそこに偶然による「ゆらぎ」を足して出力を得る，という感じです。

春：この図1のような手法が，第1世代のシミュレーションゲームには使われているというわけね。

Ats：さて，ここでキーワードとして取り上げたいのが「フィードバック」という言葉なんです。

M：フィードバックというと？

Ats：たとえば，株式投資のシミュレーションがあるとしますよね。この場合，手持ちの資金を株に投資，つまり入力とし，お金を出力として得る。

護：そして投資の結果資金が増えれば，当然次回の投資は大きくできる。

Ats：このように，出力によって入力が変化するような仕組みをフィードバックと呼ぶんですが，さっき紹介した「医は算術なり」というゲームには，このフィードバックという仕組みが組み込まれていなかったんです。

春：つまり，お金を稼いでも病院を拡張したりできなかった，ということ？

M：で，このフィードバックの有無が，第1世代と第2世代の境目になると……。

Ats：そんなふうに早とちりしないでくださいよ。確かに，フィードバックの有無は

ゲームの面白さを左右する大きな要素ではあるんです。でも，それがすべてかというとそうでもない。
護：それはどのような意味でしょう。
Ats：さっきもいいましたけど，経営シミュレーションのゲーム中で使われている計算って，それほど複雑なものではないんです。専門用語で「重回帰モデル」と呼ばれている，各入力に特定の係数をかけてすべてを加える，という単純なモデルが使われている場合がほとんどです。
護：なるほど。すると，資金を増やすために有用な方法，つまり掛け合わされる係数のいちばん重要な要素が，しばらくゲームをやっているとおのずと見えてくるというわけですね。
春：具体的にいうと？
M：「信長の野望」で，開墾をして年貢を増やすより穀物を取り引きしたほうがお金が貯まる，とかね。
Ats：やっぱりゲームって，内部の仕組みがわかってしまうととたんに面白味がなくなるんですよ。プログラミングされた数式が想像できてしまうと，いままで感情移入していたものが急に色あせてしまうっていうか……。
春：あ，それってなんとなくわかる。
Ats：ゲームが内部にどれだけ複雑なシステムを持っているか，ということは面白さを決定する大きな要因になると思うんです。
護：しかし，ただ複雑さを増すだけなら，ゲームで扱う要素の数を増やせばいいのではないでしょうか。
Ats：それがまた問題なんですよ。ゲームの仕組みがあまりごてごてしてくると，今度は取っつきやすさを損ねてしまいます。
護：シミュレーションのシステムを複雑にすればゲームは面白くなるが，実際に複雑にしてみると，とっつきにくくなるというわけですか。ならば，面白い社会科学系シミュレーションゲームを作るのは不可能，ということになってしまいます。
Ats：そんなことないですよ。現実に，「シムシティー」みたいなゲームが登場してきたのが最もいい反証になると思います。

そして第2世代へ

M：「シムシティー」って，内部ではけっこう複雑なことをやっているように見えますよね。
Ats：でも，シミュレーションの基本になっている部分は，図1のような方法論なんですよ。
M：そうかなあ，まったく別次元のことをしているように見えますけどねえ。
Ats：うーん，なんていえばわかってもらえるかなあ。「シムシティー」って，ゲーム画面が格子状に区切られているでしょう。
春：その格子の上に道路を引いたり区画整理したりして，町を育てていくのよね。
Ats：で，その格子1つひとつが図1でいう入力になっているというか……。言葉だけじゃわかりにくいから，こちらも図にしてみましょう（図2）。
M：なるほど。この図を見る限りでは，シミュレーションの基本となる部分は第1世代のゲームと変わらないようですね。
春：ちょっと待って。確かさっきの話だと，図1の「入力」の部分は，すべてプレイヤー側から指示される，っていうようなニュアンスがあったけど。
護：確かにそうでした。事実，経営シミュレーションゲームを見ると，なにをいくら投資をするか，どのような行動指針をとるかなどをプレイヤーが指示し，天変地異などの偶発的な要素が加えられて結果が決まる，というものがほとんどです。
春：だけど「シムシティー」って，プレイヤーがなにも行動を起こさなくても，勝手に状況が変化していくじゃない。市街地を放置すると，スラム化して人口密度が減っていくとか。
M：そういえばそうですね。スラム化というのは，図1にあるような「偶発的要素」というよりはむしろ，必然的なできごとといったほうが的を射ていますよね。
護：つまり人間の側が特定の入力作業を行わなくても，計算時の「入力」が勝手に変化していってしまう，ということになります。このようなシステムはどのように捉えればいいのでしょう。
Ats：それはですね，格子状に分けられた要素同士が，互いに影響し合うような仕組みになっている，と解釈できると思います。つまり，入力同士がフィードバック系を成している，という……。
春：ひ～，わけがわからない。
Ats：スラム化を例にとるとしましょう。
1）　公害が少なく，交通の便がいい住宅地区は地価が上がる
2）　地価が上がると高層化して人口密度が増える
3）　人口密度が増えると悪者も増える（スラム化する）
4）　スラム化すると地価が下がる
M：そして，地価が下がると人口密度が減る，というわけですね。
Ats：とにかく，こんな具合に，マップ上のできごとは進んでいくわけです。
護：実際は，地価や人口密度などは数値パ

図1　経営シミュレーションの基本

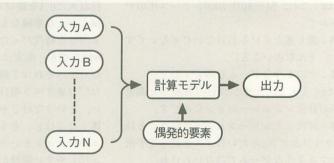
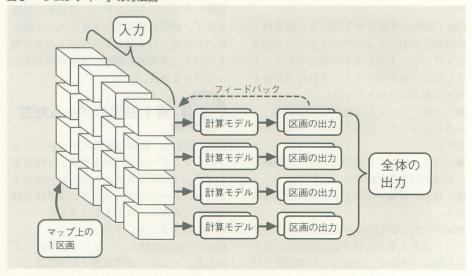



図2　「シムシティー」の方法論

ラメータとして記憶されているのでしょうから，数値が一定の範囲を越えたらスラム化する，などという処理の流れになるのでしょう。

春：これがフィードバックなわけ？

Ats：いま挙げた4つのできごとを図に描いてみるとすると，ちょうど輪の形になるじゃないですか。これって，今回の始めのほうで話した「得たお金を次回の投資に回す」というのと同じでしょう。

春：なるほど。

M：川の水も海に流れるばかりじゃすぐに枯れてしまうけど，水が蒸発して雲になって上流で雨が降るから，いつまでも水が流れている，というようなことですね。

護：すると，ゲームのシステムを複雑にしつつ，取っつきやすさを失わない工夫が，この「入力同士をフィードバックさせる」ことにあるということでしょうか。

Ats：そう，僕のいいたかったのはそれなんですよ。つまり「シムシティー」ではシミュレーションの際に扱う要素数は従来のゲームに比べて爆発的に増えているのだけれど，プレイヤーに要求される操作はそれほど複雑化していない。繁雑な部分は，パラメータ同士をフィードバックさせることでコンピュータが自動的にやってくれるようになっているんです。

M：そして，この点が第1世代と第2世代を分かつものであると。

Ats：まあ，そんなところです。

そしていつものごとく……

Ats：えー，実はですね。突然こんな話を持ち出したのはほかでもない，これからしばらくの間，社会科学系のシミュレーションゲームを作るためのノウハウみたいなものを探っていこうと思っているんです。

護：しかし，それはかなり大変なのではないでしょうか。先ほどの話のように，マップ上の格子ごとに計算モデルを適用してシミュレーションを行うという大まかな方法論はいいとして。

春：それ以外に難しいものって，どんなことがあるのかしら？

護：格子で区切られた入力要素がそれぞれフィードバック系でつながれているようなシステムを扱うとなると，計算量は指数関数的に増大して手に負えなくなるような気がしますが。

Ats：確かに，くそ真面目に計算をするとなるとそういうことになるでしょうね。だけど，そこはゲームなんだから，はしょれる部分はおもいっきりはしょってしまえばいいんです。「シムシティー」にしたって，都市の人口は増減するけれど，その人たちがどこから来て，どこに行くのかはまったく考慮されていないじゃないですか。

M：そういえば，マップ内の人の移動もそれほど厳密に再現されているわけではなかったような気がしますね。

Ats：まあ，はしょるとはいってもさすがに計算量は大きくなるでしょうから，無駄を極力省くようなアルゴリズムを検討してみるつもりでいます。

M：ところで，社会科学系のシミュレーションゲームの世代の話の続きなんですけど。

春：そうそう。第1，第2ときたんだから，第3世代もあるんでしょ？

M：「タワー」とか「テーマパーク」とか，新しいゲームも出てきていますしね。

Ats：僕が見たところ，第3世代と呼べるようなシステムを持ったゲームはまだ現れていません。「タワー」なんかは，シミュレーションのシステムとして見ると「シムシティー2000」よりも簡単な気がします。ただ，エレベーターの管理，という題材には大変魅力がありますけど。

護：あの，先ほどの世代区分を見ていてふと思ったのですが。

M：なんですか？

護：第1世代のシステムは入力から出力へのつながりが1次元的，そして第2世代は2次元的といえないでしょうか。

M：なるほど。確かに，第1世代は計算方法が直線的だけど，第2世代は平面的な広がりを持っているように見えますね。

春：すると，第3世代はマップが3次元的に展開するってこと？

Ats：それはどうかなあ。だいいち，素材として3次元的なマップを使うようなものが見つかるかどうか，非常に疑問ですよ。

M：「シムシティー」の街が階層を成していて，各階がエレベーターでつながっているっていうのはどうですか？

Ats：マップが3次元になったって，上下の階層がフィードバックのような仕組みで結ばれていなければダメなんですよ。

春：じゃあ，柴田君は第3世代にはどんなゲームがふさわしいと思うの？

Ats：それは……。

春：それは？

Ats：それがわかれば苦労はしません！

M：やっぱり。そうくると思ったんだ。

護：アマいですね。あなたまさか，そのセリフを今月のオチにしようと思っていたのではないでしょうね。

Ats：……。

M：あっ。いい返せないところをみると，どうやら図星らしいですよ。

Ats：チクショウ，あんたいつもひと言多いんだっ！

M：ちょ，ちょっと2人とも，事務所でケンカは困りますよ。

護：ふっふっふ。こんなこともあろうかと出番のないこの数カ月，ひそかに中国拳法の修行をしていたのです。さあかかってきなさい，ちぇえすとお！

Ats：くそうコシャクな，キエー！

春：きゃー，2人ともステキだけど怪我だけはしないでね。

M：すいませんねえ，いつもこんな終わり方で。

（つづく）

アンドゥが使えない日々

Takazawa Kyoko
高沢 恭子

Oh!Xにもいつもパソコンを使ったイラストを描いてくれているキョウコさんのところに，童話の挿絵の依頼がきました。と，ここまでが先月号。今月はその続きでイラストレーターの仕事のお話です。

いっしょに走ろう

EPSONのキーボードをバッグにつめて，三重から帰京した。

「キュウハチのキーボードは，リターンキーがよく戻らないよ」と，以前からトオルが訴えていた。リターンキーがノーリターンキーになったという。

NECのPC-9801VMは，使いはじめて7年目くらいだろうか。三重にあるDOS/Vパソコンとの時代差を痛感するといいたいところだけれど，有能ぶらずにこじんまりとしごとをこなす明解なマシンには，捨て難いものがある。

家じゅうで叩きつづけたキーボード。なかでも打撃の8割を一身に受けているといってもいいリターンキーが，いちばん先にリタイアするのは道理である。

「だんだん歪んできて，垂直の上下運動ができなくなったのかしら」といっているうちに，「こんどはカーソルキーもおかしいよ」となった。やはり，はたらき者のキーから順に故障していくらしい。

「EPSONはキュウハチにつなげられるからちょうどよかった。これ新品なのに向こうで遊んでたのよ」

キーボードは更新されたけれど，トオルはあまり喜ばない。以前から「学校のパソコンはいいよ」「プリンタは音なんかしないよ」と，それとなくわが家の旧式システムを指摘していたのに，こんども改革のチャンスではなかったのだから。

先日も，大学の友人がつくったという小さな冊子をもってきた。「ジーザス・クライスト・スーパースター」の歌詞を自分で翻訳したものだそうだ。「山本君はマックでこれをこしらえたんだよ。きれいでしょぉ」という。「レーザープリンタがきれいなんでしょ。そのうち買うからね」なんて，私も強がりの出まかせをいう。

じつはX68000のハードディスクも数カ月前から動かなくなった。もっぱらフロッピー起動ではたらいている。功労賞もののプリンタも疲れがみえて，用紙の送りでトラブルが頻発している。三重のマシン軍団にくらべると，東京の家の機器は衰弱が目立つ。

ふだんは聞き流している新マシンのウワサ話も，こうなってくるとおおいに気にかかる。でもいちどくらいは，古いなじみのパソコンが，ほんとにパタッと止まる日まで，いっしょに走ってみるのもいいじゃないかと思う。

ことわれなかった話

亡くなられたご主人が残した童話の原稿を本にしたいので，挿絵を描いてほしい。数カ月前にこのお話があった。

68歳で病死されたご本人は，羽曽部忠(はそべ・ただし)さんという児童文学者で，いまも小学校の国語教科書に詩の作品が多数掲載されているそうだ。

羽曽部さんは新宿区内で小学校の先生をしながら童謡や詩を作り，作品集を何冊も発表した。新美南吉児童文学賞をはじめ，いくつもの受賞がある。その教員時代に新宿のおばあちゃんと10年間ほど同僚だったのだ。

いつかおばあちゃんが，自分の海外旅行の記録を本にして知人にさしあげたことがあり，私がイラストを描いた。贈呈本が熱心に読まれることはマレと思うのに，羽曽部ご夫妻はその本を細部までくりかえし読んでくださり，とくに絵を気に入っておられたのが，亡くなられた羽曽部さんご本人だったという。

そんなご縁から，今回ぜひ，最後になった本の挿絵をとたのまれたのだから，光栄なことではあった。

400字詰の原稿用紙で180枚ほどの手書き原稿のコピーが，さっそく送られてきた。「話を読んで，気がすすまなかったら，むりをしないでください」。ご自身も小学校の先生だった夫人は，そういってくださった。

はじめの難関は180枚の原稿を読むことだった。じょうずな文字だが少なからず乱れもあるし，複雑な挿入，削除，訂正はまさしく純粋のナマ原稿である。けっして読みやすいものではない。すこし前まではみんなこうだったのに，遠い時代の遺産を目にするようだ。

話の内容も手書きの原稿にふさわしいものといったらいいだろうか。大きなけやきの木の下に清水の湧く過疎の村。数軒しかない家と小さな分教場。そこで大人とこどもたちが，いっしょうけんめい大切なふるさとと自然を守っていく。それが大きなテーマだった。

長編であるという重さと，古風なふんいきに少々とまどった。知らない風習も登場する。たいせつな最後の作品であるというのに，いいかげんな絵を入れてだいなしにしてしまうかもしれない。

心配になって時代設定をたずねた。モデルとなった舞台はご夫妻の郷里である福島県会津市の山村で，時代は高度経済成長期に入っていた昭和50年代を想定していたらしいということだった。それならトオルが誕生して以後の時代であり，そんなに昔の話にあるような貧しい山村のようすを描かなくてもよい。

気がかりがもうひとつあった。物語の随所にあらわれる文明批判を，いちばん浴び

なければならないのが，いまの私の日常生活であることだ。

決心しかねるまま，東京で夫人におあいした。生前お目にかかれなかった羽曽部さんのお人柄や作品，ご夫妻で私の絵をいかに気に入っておられたかなどの話を，いろいろとうかがった。母の本にあった私の絵に，1枚ごとにくわしく感想をのべられたのには恐縮してしまった。

私はお引き受けするしかないという気持ちになって，あらためて自分の状況をお話しした。パソコンやワープロによる日常を送ってしまっていること。イラストさえもほとんど描画ソフトを使うこのごろであること。持参したパソコンによる絵もお見せした。

そしておたずねした。

「挿絵は筆と絵の具で描くつもりですが，こういう立場にいる者の絵でもよろしいのでしょうか」

「なんの不服もありません。絵についての注文もまったくありません。自由に描いていただきたいんです」

迷いながらもお引き受けした。

1枚目がこわい

はじめの作業は，物語を読みなおしながら，章ごとに内容を要約し一覧にすることだった。文章の量の配分を知るために，原稿の枚数を入れ，季節もメモした。

つぎは人物に関するメモづくり。年齢や学年，しごと，性質，エピソードなど。5軒ほどの家の家族構成も図にした。

それから，季節をあらわす花や植物，樹木の名前。方言として使われている特殊な言葉の由来や意味なども整理した。

これらはみんなパソコンでおこない，必要に応じて書き加えもしていった。この資料は最後までしごとの進行を助けてくれることになった。とくに家族構成の図は，固有名詞を頭に入れ，キャラクターをつくるうえでおおいに役だった。

こうして，作者がつくった世界のなかの条件をひとつでも多くあつめ，それを手がかりに，作者の心を伝えるような情景を自分なりに表現しなければならない。

絵はすべて縦サイズで，ページいっぱいに描くよういわれていた。

1枚目にどの場面を選んで描くかで，全体が決定するだろう。そう思って，いちばんやっかいと思われるシーンを選んだ。

村から町へ引っ越してしまう女の子の一家のために，お別れの『賽（さい）の神の祭』をやる場面だ。もう村では何年もおこなわれていない行事を，大人たちがやって見せるのだ。

illustration：Kyoko Takazawa

本文によると，どうも転勤先で体験したお正月の行事「どんど焼き」と似たものらしい。しめなわやワラ，ダルマなど，お正月の縁起ものをうず高く積みあげて燃やすのだ。たしかトオルが小学校3年くらいのときのビデオがあった。みつけだして再生してみたが，物語のなかにあるような小道具は見あたらない。

本文には「まるで魚釣りをするようにモチ焼き網を竹の枝の先にさげて」その上におモチをのせて焼くのだとある。およそのようすはわかるけれど，会津地方の行事として正確に描こうとしたら，これだけの説明ではムリがある。

けっきょく羽曽部夫人に電話して教えていただいた。モチ焼き網の四隅にハリガネをくくりつけ，まんなかにピラミッド型にあつめて，枝葉をすべて切り落とした竹の先につり下げるのだそうだ。この火で焼いたおモチを食べて，健康を願う。

こどもたちのリーダー格である晋平君にこの竹を高々とかかげさせて，画面の手前にこちら向きに立たせた。その隣に町へ降りていってしまうマツ子ちゃんを並べた。背後いっぱいに，祭のふんいきがわかるよう「夜空をこがす」炎を囲む村人たちを遠景で描いた。

ワザがあらわれるまで

1枚目の絵は，イメージに近いものがなんとかできた。やはり，じっさいに筆をとって紙に描くことは，満足感がちがうものだという思いもあった。

1枚の絵のためにたくさんのスケッチと下絵をつくる。同じ人物がつぎの場面でちがうキャラクターになってしまわないように，頭のなかで人物を動かしたり，架空のカメラアングルを変えてみたりする。

人物をはじめ，衣服も風景も建物も，ぜんぶ自分がつくったものを配置していいのだから，長い物語をたったひとりで映画化しているようなものだ。責任は重大だが，こんなおもしろいことはない。描きすすめるうち，人物たちが絵のなかで自立していくのも実感できた。

つぎつぎに11の場面を絵にしていったが，1場面のために平均3枚くらいのやりなおしをした。

筆で描く絵は1点かぎりで，やりなおしがきかないところに値打ちがある。しかし予期しない絵ができたら，用紙をあらたにして，はじめから描きなおさなくてはならない。瞬間芸のようなワザを見せなくてはならないから，いいワザがあらわれるまで時間がかかる。パソコンならできるアンドゥのコマンドがないのだ。

難関の表紙カバーの絵はこれからだが，おかげで数カ月間，あらためて絵筆を使う研修ができた。

©KYOKO EGUCHI
VOL.40
Another
CGわ～るど
ことしも
よろしくね～

お正月は・・

みんなおやすみ・・
1月 1995
1
さあ1年のはじまりだ～

マシンもお休み・・
休

そして人間も・・
ぐおー
やっぱり
寝正月に
かぎる…

美少女に
ヘンシンして
ドンペリニヨン
(超高級シャンパン)
の酒ブロに
はいって
いる…
しあわせ
なひと

ゆめは
いちばん
てがるな
バーチャル
リアリティー
である…

うんにゃ
?
いいとこ
だった
のに

初もうでかぁ～ ちとめんどくさいが
1年にいっぺんのことだから
いくかね～～

ず〜
さぶ〜
あ
おさいせん
忘れた
！
ゆめ ゆめ
ゆめよ
でてこい
DREAM
COME
TRUE
ちょい
まち
いまこしら
えるから
???
はいっ
バーチャル
マネー！
ここからひだりはゆめです…
しんじられないなー
なんでも INTERNET
には、その中だけで
使えるお金つーもん
があるそーな…

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

DSP高速演算プロセッサボード
AWESOME-X
グラビス

AWESOME-X

グラビスはX680x0用DSP高速演算プロセッサボード「AWESOME-X」を発売する。

同ボードはDSP(Digital Signal Processor)にTMS320C26B-40MHzを採用し,DSP用のワークとして64Kバイトの RAMを装備している。接続端子は,最高128,000bpsまで対応したRS-232C 2系統,DAI(光Digital Audio I/F)1系統などが用意されている。

付属されるソフトウェアは,FLOAT2.X互換ドライバ,DSP直接制御FLOATドライバ,高速シリアルドライバ(~128,000bps),シリアルMIDIドライバ,JPEGデコーダ/エンコーダなどが予定されている。

また,オプションとしてMIDIドーターボードや赤外線通信ユニットなどが発売される予定。

価格は89,800円(税別)。

〈問い合わせ先〉

㈲グラビス ☎044(812)7499

携帯情報ツール
PI-5000/FX/DA
シャープ

PI-5000

シャープは“ザウルス”に新開発の通信ソフト“ザウルスネット”を搭載した“アクセスザウルス”「PI-5000」「PI-5000FX」「PI-5000DA」を発売する。

「PI-5000」はオプションのFAXモデム「CE-FM3」や携帯電話アダプタ「CE-DAI」を接続することで,各種パソコン通信へのアクセスが容易になった。特にNIFTY-Serveへのアクセスは,富士通とのソフト共同開発により画面のアイコンをタッチするだけで行える。

また,従来機「PI-4000」に比べ表示や編集スピードが1.5倍にアップした。そして,国語,漢和,英和,和英の4冊の辞典を内蔵し,相互関連検索や作成文書への取り込みも可能になった。また,ビジネスソフトやゲームなどのプログラムをパソコンから読み込んでアプリケーションを追加できるアドイン機能も搭載している。記憶容量は従来機種の約2倍の1Mバイトでユーザーエリアは約700Kバイト。

「PI-5000FX」は「PI-5000」とFAXモデム「CE-FM3」がセットになったアナログ回線用データ通信/FAX送信システム。そして,「PI-5000DA」は「PI-5000」と携帯電話アダプタ「CE-DAI」がセットになったデジタル携帯電話用データ通信/FAX送信システムである。

価格は「PI-5000」が82,000円,「PI-5000FX」が98,000円,「PI-5000DA」が141,000円(それぞれ税別)。

〈問い合わせ先〉

シャープ㈱ ☎06(621)1221, 03(5261)7271

マッハジェットプリンタ
MJ-450/MJ-1050/MJ-1050V2
セイコーエプソン

MJ-450

セイコーエプソンはマッハジェットプリンタ「MJ-450」「MJ-1050」「MJ-1050V2」3機種を発売した。

「MJ-450」の解像度は360dpiで,印字速度は110cps(全角)。横縞による印字ムラはマイクロウィーブ機能により解消された。中間色の表現もモノクロハーフトーン出力機能により美しく再現できる。さらに,縮小印刷モードとして80%と50%の2種類が用意されている。インクカートリッジのランニングコストは1枚(英数仮名1500文字/A4)あたり約3.2円。オートシートフィーダを標準で装備しており,A4用紙なら100枚までの連続印刷ができる。また,手差し給紙も可能。

「MJ-1050」は,印刷用紙がA3サイズまで対応している。印字モードは2種類あり,高速印字モードでは167cps(全角)を実現した。プリンタ制御コードはESC/Pを搭載。さらに,オプションで連続紙にも対応できる。

「MJ-1050V2」は「MJ-1050」の機能に加えて,連続紙が使えるトラクタユニットを標準装備し,帳票やラベル印刷に対応できる。また,プリンタ制御コードにESC/P V.2を搭載している。

価格は「MJ-450」が39,800円,「MJ-1050」が79,800円,「MJ-1050V2」が99,800円(それぞれ税別)。

〈問い合わせ先〉

エプソンインフォメーションセンター

☎0424(99)7133, 06(399)1115

CD-ROMドライブ
PX-45CS/PX-43CS
プレクスター

プレクスターは4倍速CD-ROMドライブ「PX-45CS」(外付け型)と「PX-43CS」(内蔵型)を発売した。

同機の特徴は以下のとおり。

転送レート:614Kバイト/sec

PX-43CS/PX-45CS

アクセス速度：
　160ms（ランダムアクセス時）
　125ms（ランダムシーク時）
データバッファ容量：256Kバイト
インタフェイス：SCSI2対応。
価格は「PX-45CS」が49,800円で「PX-43CS」が44,800円（それぞれ税別）。
また，1993年7月に発売された同社の4倍速CD-ROMドライブ「PX-45CH」が69,800円，「PX-43CH」が59,800円（それぞれ税別）に値下げされた。
〈問い合わせ先〉
プレクスター㈱　☎03(3847)8281

液晶ビジョン
XV-P3
シャープ

XV-P3

シャープは液晶ビジョン「XV-P3」を発売した。
同機は，液晶パネルにカラーフィルタを使用しない単板式光学システムを採用し，コンパクトで従来機に比べ約4倍の照度を実現した。液晶パネルのサイズは3.6型で画素数は301,158画素。また，投写距離が2.8mで80型の大画面が楽しめる。
接続端子はS映像入力1系統，ビデオ入力2系統，ヘッドフォン出力1系統など。また，最大3Wのアンプとスピーカーを内蔵している。
大きさは355mm(幅)×341mm(奥行)×201mm(高さ)で，重さが約6kg。キャリングハンドルつきで持ち運びも可能。
価格は220,000円（税別）。
〈問い合わせ先〉
シャープ㈱　☎06(621)1221，03(5261)7271

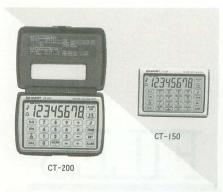
パーソナル情報ツール
PA-B1
シャープ

PA-B1

シャープは機能を限定したコンパクトな“ザウルス”といった感じの小型パーソナル情報ツール「PA-B1」を発売した。
同機はディスプレイに電話帳やスケジュールなどが一覧できる（漢字8文字×5行，横96×縦64ドット）大型液晶パネルを採用し，そこに表示された各種キーボードやアイコンをペンでタッチするだけで簡単に操作できる。
主な機能として，電話帳やスケジュール，メモ，電卓，世界時計，各種の予定を優先順に並べ換えてくれるアクションリストなどの機能を搭載している。また，ひとつの情報からそれに関連する情報を検索する関連検索を行うことが可能。
記憶容量は32Kバイトで，そのうちユーザーエリアが約24Kバイト。大きさが110mm(幅)×80mm(奥行)×13mm(厚さ)で，重さは約108gと非常にコンパクトで軽い。
価格は11,000円（税別）。
〈問い合わせ先〉
シャープ㈱　☎06(621)1221，03(5261)7271

時計付電卓
CT-150/CT-200
シャープ

シャープは“ザウルス”に取り入れた液晶タッチパネルをディスプレイに採用した，キーのない時計つき電卓「CT-150」「CT-200」を発売した。
「CT-150」は通常の電卓，時計機能以外に，世界16都市の世界時計機能をもつ。
「CT-200」は世界時計機能はないが，ハードカバーがついており持ち運びのときに安心できる。

CT-200　CT-150

価格は「CT-150」が2,700円で，「CT-200」が4,000円（それぞれ税別）。
〈問い合わせ先〉
シャープ㈱　☎06(621)1221，03(5261)7271

INFORMATION

デジタル・エンターテイメント・グランプリ'95
ケイネット

ビジュアルパソコン通信のケイネットでは「コミック」と「マルチメディア」をテーマにした娯楽作品のコンテストを東映ビデオ株式会社の協賛で開催する。コンテストの内容は以下のとおり。
応募締め切り：1995年3月末日
応募テーマ：
・「オンライン・コミックコンテスト」
デジタルデータによるコミック作品。1枚の絵だけでもよいし，画面をクリックするとストーリーが展開するパターンなど，形態は自由
・「マルチメディア・タイトル制作レース」
「CD-ROMタイトルにして売り出すことを前提とした」作品企画
応募条件：応募作品は作者本人による未発表のオリジナル作品
審査方法：パソコン通信(K-NET)上の特設コーナー「DEG'95サーキット」にすべての作品を公開しK-NET会員の各作品のダウンロード回数を集計し，審査員が協議のうえ決定する予定
賞金：グランプリ100万円ほか
応募方法：作品をK-NETにアップロードするか郵送するかで行う。作品はデジタルデータに限り，フォーマットはQuickTimeムービー，JPEG, BMP, TIFF, PIC, PICT形式など。プラットフォーム，動作環境をできる限り詳しく書くこと。応募要項の詳細は下記の電話番号に
〈問い合わせ先〉
ケイネット「DEG'95」係　☎045(633)5112

FILES Oh!X

このインデックスは，タイトル，注記──著者名，誌名，月号，ページで構成されています。あけましておめでとうございます。いよいよ1995年に突入です。なにか目標を決めて新しいことに挑戦してみませんか？

参考文献
I/O　工学社
ASAHIパソコン　朝日新聞社
ASCII　アスキー
コンプティーク　角川書店
C Magazine　ソフトバンク
電撃王　主婦の友社
PIXEL　図形処理情報センター
マイコンBASIC Magazine　電波新聞社
My Computer Magazine　電波新聞社
LOGIN　アスキー

一般

▶フリーソフトウェア＆シェアウェア33観音
Windows, MS-DOS, Macintoshで人気を集めているフリーソフトウェア，シェアウェア計33本を紹介。──編集部，ASAHIパソコン，11･15号，20-31pp.

▶98ユーザーのためのマッキントッシュ教室　2
PC-9801とMacintoshの違いに焦点を当て，それぞれの設計思想を解説する。お題は「画面にＡ＞が出てこない」。──荻窪圭，ASAHIパソコン，11･15号，32-35pp.

▶EDUCATION
清泉女学院中学・高等学校では，Macintosh23台を導入している。レポート課題のツールとして利用する。──坂本伸之，ASAHIパソコン，11･15号，44-45pp.

▶ハードウェアFLASH！
ソニーのMDデータドライブ発売などのニュース。──編集部，LOGIN，22号，40-43pp.

▶Amusement Paradise
SEGA SATURNの特集。周辺機器とソフトのラインナップを紹介する。また「'94アミューズメントマシンショー」と「第15回CSG新作発表会」のレポートも掲載。──編集部，LOGIN，22号，190-197pp.

▶架想楽園へ行こう　Ver.2.04
「バーチャルブルワリーアドベンチャーの巻」と題して，恵比寿ガーデンプレイスにある「恵比寿麦酒記念館」を訪問する。──中田宏之，LOGIN，22号，212-215pp.

▶NEWS COLLECTORS
任天堂の32ビット機「バーチャルボーイ」の詳細をメーカーに直接インタビュー。ほかに，松下電器産業の大容量メディア「PD」の情報など。──編集部，電撃王，12月号，20-23pp.

▶これなら買い！　SEGA SATURN
SEGE SATURNのソフトや周辺機器を紹介し，今後の将来性を見る。PlayStation, PC-FXの情報も掲載。──編集部，電撃王，12月号，24-42pp.

▶香港の謎と真実
香港のパソコンソフト，ハードの販売事情など。──編集部，電撃王，12月号，62-65pp.

▶ゲーム業界就職必勝マニュアル
グラフィックデザイナー，プランナーなどへのインタビューからゲーム業界の魅力を探る。また，ゲームスクールの現状と将来に触れる。──編集部，電撃王，12月号，141-150pp.

▶特集!!　次世代機の登場でゲームはこう変わる！
SEGA SATURN, PlayStationを中心に，ソフトの陣容，ソフトハウスの次世代機への期待などを掲載。──編集部，コンプティーク，12月号，21-31pp.

▶趣味の基板IIX　SPECIAL
渋谷パルコにて行われたテレビゲームの名作展覧会，「テレビゲーム・クラシックス」の模様をレポートする。──編集部，コンプティーク，12月号，124-127pp.

▶超級電脳遊戯中心　FINAL STAGE
過去のマイナー作品，カルトといわれるゲームを特集。覆面座談会で各方面のゲームが続々登場。──編集部，コンプティーク，12月号，138-139pp.

▶大容量化する記憶媒体たち　第２回
光磁気ディスクを紹介する。光メディアや磁気メディアとの違いを解説。──編集部，マイコンBASIC Magazine，12月号，44-45pp.

▶Arcade Game Graffiti
アーケードゲームの歴史を振り返るシリーズ，1981年のPart３。スクロールゲームと固定画面ゲームの傑作を紹介する。──編集部，マイコンBASIC Magazine，12月号，166-169pp.

▶パソコン買うならペンティアム
種類も増えて値段も下がってきたPentiumマシンのなかから，６機種を選んで徹底比較する。──編集部，ASAHIパソコン，12･1号，16-29pp.

▶98ユーザーのためのマッキントッシュ教室３
「ファイル名は８文字に縛られない」と題して，PC-9801とMacintoshのファイルシステムの違いを考察する。──荻窪圭，ASAHIパソコン，12･1号，30-33pp.

▶アメリカ・マルチメディア報告
「情報スーパーハイウェーの中身はどこにある？」と題して，マルチメディア運営側と一般ユーザーの間にあるギャップについて述べる。──高間剛典，ASAHIパソコン，12･1号，162-163pp.

▶ハードウェアFLASH！
アイ・オー・データ機器から発売された低価格のCCDカメラGV-CDC，富士通FM TOWNSの新ラインナップ紹介など。──編集部，LOGIN，23号，42-45pp.

▶THE NEWS FILE
PlayStationの発売日決定，PC-FXの詳細，Silicon Graphics EXPO'94のレポートなど。──編集部，LOGIN，23号，46-51pp.

▶特集　ゲーム道
カルトなゲームの遊び方を特集する。──編集部，LOGIN，23号，171-185pp.

▶Hardware Forum X
マルチメディアの必需品，外付けスピーカーの特集。各社の製品を紹介し，セッティングのポイントなどを教える。──編集部，LOGIN，23号，186-189pp.

▶ハイテクCGオペラ
アメリカのCGパフォーマンス集団のジョージ・コーツ・パフォーマンス・ワークスを取材する。──編集部，LOGIN，23号，218-221pp.

▶この冬の最新主要機種
「98MULTi CanBe」「PC-586RA/RV」「COLPO 33」など，各社から発売された新製品を速度テストも交えて紹介する。──編集部，ASCII，12月号，317-340pp.

▶これが戦略型年賀状だ！
年賀状ソフトの集中レビューと，年賀状印字に適したプリンタ選び，プリントゴッコの活用法など。──編集部，ASCII，12月号，349-372pp.

▶新科学対話　最終回
千葉大学文学部助教授の大澤真幸氏を迎えて，新メディアの普及が人間の社会に対する規範をどう変えるかについて対談する。──竹内郁雄，ASCII，12月号，404-410pp.

▶魅惑のニューテクノロジー　第９回
PCMCIAカードの規格や特長を解説する。──編集部，ASCII，12月号，422-427pp.

▶DIGITAL WATCH　第11回
ラピドシステムズが始めた個人向けインターネットアクセスサービスなどを題材に，ネットメディアの可能性について対談する。──桝山寛＋David D'Helly，ASCII，12月号，432-435pp.

▶「手帳」を考える'95
ベストセラー「「超」整理法」の著者である野口悠紀雄氏に，その整理法の内容と，氏が開発した「超」整理手帳についてインタビュー。──編集部，ASCII，12月号，459-463pp.

▶稀代もののけ考
面白いグッズを集めて紹介するページ。今回はノンジャンル。ライトつきルーペ，GPSレシーバなど。──編集部，ASCII，12月号，486-487pp.

▶こんなに使える著作権フリーの「素材集」
各社から発売されている著作権フリーのCD-ROMを紹介する。──矢野光一，I/O，12月号，49-51pp.

▶動画圧縮技術「MPEG」のすべて
データ圧縮規格「MPEG」の内容はどうなっているのか，なぜ映像がなめらかなのか，符号化の技法などを解説。──矢野光一，I/O，12月号，69-72pp.

▶激安　低価格カラープリンタ・ガイド
10万円を切ったカラーインクジェットプリンタと，値段を下げつつある昇華型プリンタを紹介する。──編集部，I/O，12月号，85-92pp.

▶高解像度ディスプレイ
最新のディスプレイ製品紹介と，失敗しないディスプレイ選びのポイント，ディスプレイの調整法など。──編集部，I/O，12月号，93-100pp.

▶ワイヤレス・コンピューティング時代がやってくる！
今後のワイヤレス技術を予測する。──ローレンス・J・マギッド，I/O，12月号，104-110pp.

▶光磁気ディスク「MO」で広げるパソコンの実践的活用(1)
各種の記録メディアのなかに見る光磁気ディスクの優

位性，ディスクの種類などを解説。──佐田守弘，My Computer Magazine，12月号，44-47pp.

▶パソコン研究室

パソコン大掃除法を伝授する。──Space Club，My Computer Magazine，12月号，127-129pp.

X1/turbo/Z

X1シリーズ

▶「なぬっ!?」

2人用対戦型ゲーム。相手よりも早く，指示されたものを見つけると勝ち。──ちひろちゃん，マイコンBASIC Magazine，12月号，99-100pp.

X68000

▶Release Data

発売予定のゲームを機種別に紹介。X68000用「VIEW POINT」など。──編集部，LOGIN，22号，8-9pp.

▶NEW SOFT

X68000用はネクサスインターラクトの「VIEW POINT」を紹介。──編集部，LOGIN，22号，10-25pp.

▶ニューソフトショー'94

年末年始の新作ゲームを紹介する。X68000用「バックランド」など。──編集部，LOGIN，22号，147-179pp.

▶未確認クリエイターズ

読者投稿のゲームプログラムを紹介するコーナー。シューティングゲーム「ばろろぎす」とシミュレーションゲーム「ダンジョン　マネージメント2」の，X68000用2作品を掲載。──編集部，LOGIN，22号，198-201pp.

▶電撃王掲載全ゲームインデックス

X68000用「上海　万里の長城」ほか，11月発売予定のゲームを紹介。──編集部，電撃王，12月号，6-8pp.

▶新作王

新作ゲーム紹介のコーナー。X68000用は「バックランド」など。──編集部，電撃王，12月号，171-193pp.

▶機種別SUPER SOFT INDEX

発売予定のソフトを機種別，発売日順に掲載。X68000用「魔法大作戦」ほか。──編集部，コンプティーク，12月号，115-118pp.

▶王砦

対戦型陣取りゲーム。──コウ・アキラ，マイコンBASIC Magazine，12月号，101-102pp.

▶16 BALL

きれいな画面のブロック崩し。──渋谷正徳，マイコンBASIC Magazine，12月号，103-105pp.

▶ソーサリアン

ミュージックプログラム。消えた王様の杖のテーマ。──重長孝之，マイコンBASIC Magazine，12月号，111-113pp.

▶とびだせ！　同人野郎

X68000の同人ソフト「DIVE ON」と「過労伝説SPECIAL」を紹介。ほかに，サークルの勧誘など。──安部理一郎，マイコンBASIC Magazine，12月号，182-183pp.

▶SUPER SOFT HOT INFORMATION

X68000用は「魔法大作戦」「上海　万里の長城」など。──編集部，マイコンBASIC Magazine，12月号，とじ込み付録1-47pp.

▶NEWS&SOFT Radar!

X68000用のレンダリングソフト「XL/Image」がイマジカテクノシステムから発売されるニュースなど。──編集部，LOGIN，23号，8-9pp.

▶Release Data

11月8日現在での発売予定のゲームソフトを機種別に紹介。──編集部，LOGIN，22号，10-11pp.

▶ASCII EXPRESS

グラビスのX680x0用のDSP高速演算プロセッサボードなど各社のソフト，ハードの動き。──編集部，ASCII，12月号，285-306pp.

▶AV STRASSE

X68000用DTPソフト「XDTP SX-68K」を紹介する。アウトラインフォントを装備し，レイアウトが簡単に行える。ページプリンタ用のドライバも装備。──編集部，ASCII，12月号，445-448pp.

▶ON-LINE SOFTWARE INDEX

大手ネットにアップロードされたソフトを紹介する。X68000用JPEGローダ＆セーバ「jpeged.r」ほか。──編集部，ASCII，12月号，525-531pp.

▶なんでもQ＆A

X68000ユーザーの質問に答える。プリンタの接続が認識されないというトラブルの原因と対策ほか。──シャープ，My Computer Magazine，12月号，152-153pp.

▶NEW PRODUCTS

パーソナルリンクスのレンダラ「L/Image」のX68000版「XL/Image」ほか，ソフト，ハードの新製品のニュース。──編集部，PIXEL，12月号，36-55pp.

▶SX-WINDOWプログラミング　第14回

簡易C文法チェッカの作成。今回は，文法チェッカとSX-WINDOWとのインタフェイスとなる関数を作成する。ファイルの入力関数とエラーメッセージの出力関数など。──吉野智興，C Magazine，12月号，124-128pp.

ポケコン

PC-E500

▶10横無尽

足し算をして10の倍数を作るアクションゲーム。──アダモ，マイコンBASIC Magazine，12月号，106p.

新刊書案内

イラストで見る
インターネット入門
ジョシュア・エディングス著
村井　純監訳
鷲谷好輝訳
インプレス刊
☎03(5275)2442
AB判　177ページ
3,500円（税込）

マルチメディア本と並んで書店を賑わせているインターネット本だが，これだけ並ぶとどれを選んでどう読んだらいいのかなかなかわからない。インターネットはあまりに奥が深くて，必要な情報を過不足なく表現してくれる本などなかなかないからだ。本誌読者であれば，まず，インターネットの概念や構造を把握しないと気持ち悪いだろう。それをかったるい文章を読むことなく実現してくれるのが「イラストで読む」シリーズの最新作である本書だ。自宅のパソコンからインターネットを利用するとき，どんなデータがどういう経路を通ってどこにどうつながっているのか。これはインターネットのポイントで，なかなかイメージをつかみにくいところだが，それが詳細なカラーイラストによってややこしい文章よりずっとすばやく正確に概念を把握することができる。「TCP/IP」とはいったい何なのか，「TCP/IP」プロトコルによってどういう回線でどうつながっているのかが一発で理解できるのは（大まかなイメージだけであっても），気持ちのいいもの。しかも，具体的事例もいいところをついている。メールヘッダの読み方の解説も，インターネットがこういう構造だから，こういう情報が必要で，送信されるとこう送られていくのだ，という流れになっているから，一発でイメージが湧くのだ。

もともとアメリカで出版されたものなので，事例もアメリカ中心だが，いまの日本ならかなり似たイメージで捉えられるはずだ。パソコンユーザーが商用ネットを通して利用するケースも取り上げられている。巻末には日本のインターネットについての解説も付加されている。解説ページとイラストページがあり，解説ページはちょっと難しいが，たいていイラストページと併読すればなんとかなるだろう。まったくの初心者には難しいかもしれないが，インターネットについて知りたい人には最適の書といえよう。　　　（K）

サイエンティストになるには
生田　哲著
ぺりかん社刊
☎03(3814)8515
B6判　150ページ
1,200円（税込）

読者には理系の方が多いことと思う。そのなかでサイエンティスト，いわゆる科学者を目指している人はどの程度いるだろうか？

本書は，まさにサイエンティストになりたいという人のための本である。内容は実際に社会で活躍されている方々の体験談やサイエンティストの世界での動向や資格，待遇などについて紹介する。そして，実際になるためにどのようなステップを踏めばよいかが解説してある。

同社ではほかの職業についても出版している。就職にさえ困る世の中ではあるが，なりたい職業があれば探してみてはどうだろうか？

マルチメディア・ダス
デジタル社会研究会編
曜曜社出版刊
☎03(5259)3171
A5判　293ページ
1,800円（税込）

巷では「マルチメディア」について解説された本が溢れている。本書もそのうちの1冊である。

本書の興味深いところは，マルチメディアを探るために，現在のマルチメディアの送り手たちが自身に関わりの深いキーワードからマルチメディアについて語っているところにある。そのキーワードは，エンターテイメント，コンセプト，ビジネス，テクノロジー，クリエイティブ，カルチャー・ライフスタイルの大きく6つに分かれる。

これでマルチメディアがわかるとはいわないが，マルチメディアの広がりについて知ることはできるだろう。

先日SX-WINDOW ver.3.1のバージョンアップを受けました。

新しいシステムをハードディスクにインストールするために，ハードディスクの中身をバックアップしたあとにFORMATで装置を初期化し，新しいシステムをインストールしたところまではよかったのですが，バックアップしたファイルをリストアしていると何十枚もあるバックアップの1枚目の途中で「ファイルが読み込めません」と表示され，リストアが途中で強制的に終了されてしまいました。何度もリストアを試みたのですが，いつも同じところで終了してしまい，シャープに問い合わせてみると,「バックアップディスクを一度DISKCOPYしたものでリストアしてみる」か「正常にリストアできるバックアップディスクを用いてリストアを開始し，2枚目からバックアップディスクを変えてみる」とのアドバイスをいただいたのですが,どちらも駄目でした。バックアップを取っていなくてハードディスクがクラッシュしたのならあきらめますが，1枚目のバックアップファイルは駄目でも，2枚目以降をリストアすることはできないでしょうか？

ちなみに私の知りあいのX68000ユーザーは2人ともこの症状に泣いた経験があるそうです。全国にもこのような方がいらっしゃるのではないでしょうか？

京都府　村上　浩二

見たところ，なかなか悲惨な状況のようです。BACKUP.Xでバックアップしたファイルが壊れると，あとに続くディスクすべてに影響が及びますから。

リストア中にエラーが発生した場合は，とりあえず，

［I］無視

を押し続けることによって強制的に作業を進めていくことは可能です。これならば致命的なファイルはあきらめるとしても，ディスクの全体から見れば，はるかに軽度の損害に抑えることもできるでしょう。

しかし，ちょっと考えてみてください。問題なくそのディスクのバックアップが取れるということは「そのディスクは物理的には死んでいない」ということを意味します。書き込み時にはベリファイを行っているでしょうから，そうとうくたびれたディスクを使っていない限り，そうそう致命的な書き込み間違いは起こりそうにありません。

もちろんバックアップしたフロッピーディスク側になにか問題があった可能性も残っていますが，同じ症状の人がたくさんいるということはほかに原因があることも考えられます。

ということで結論からいうと，元のハードディスクのディレクトリ構成で，どこかに，

@IOCS
OPM
PCM

といった名前のファイルが存在していた可能性があります。こういうファイルができてしまうというのは……実にありがちなパターンですよね。

さて，ディスク上にあっても特に悪さをするわけでもなく放置されていたこれらのファイルはBACKUP.Xでバックアップされるとそのままバックアップファイルに取り込まれてしまいます。しかし，これらはバックアップされた塊からバラされてファイルになって出てくるときに，これとぶつかる名前を持ったシステムデバイスのドライバ（OPMDRV系，IOCS.X）が組み込まれていると，同名のファイルを作成できず，かといってオーバーライトもできないのでHuman68kがエラーを発生してしまうのです。

村上さんの場合もこれが原因かもしれません。とりあえず，インストールした新しいシステムから，

OPMDRV3.X
IOCS.X

あるいは，

ZMUSIC.X

を外した状態で再起動してリストアを実行してみてください（昔のシステムにあったPCMDRV.SYSは現在ではOPMDRV3.XやZMUSIC.Xに吸収されています)。

これで駄目なら今度はかなり大がかりな作業が必要になります。データ構造を解析してファイルの切り出しツールを作成せねばなりませんので。とりあえず今回はこの方法を試してみてください。

ということで今月の教訓。少々遅くても，ハードディスクのバックアップは動作のわかりやすいCOPYALL.Xで取るようにしりましょう。

C言語で半透明機能を使いたいのですがライブラリにもなく困っています。どのようにすれば使えるのか教えてください。

東京都　小澤　誠太郎

半透明機能というのはX68000のビデオコントローラが持ってる機能のうち，3種類の動作をまとめていい表した言葉です。

ひとつはグラフィック画面とテキスト＆スプライト画面の半透明合成，もうひとつはグラフィック画面のプレーン間の半透明合成，そして最後にコンピュータ画面全体と外部入力されたビデオ信号との半透明合成です。

これらは，どの領域を半透明合成するかによって最終的に7種類のモードに分かれます。これらのうち，普通使用されるのは最初の2つ，グラフィック画面を使った合成です。ここでは，ビデオ信号との合成はおいといて，その他の4つのモードについて解説します。

まず，ビデオコントローラのレジスタ2，3についてざっと理解することが必要です。レジスタ2のポートは$E82500_H$にあり，ここから2バイトの読み書きをすることで制御されます。このポートはグラフィックの各プレーン間（最大4枚）の表示順序，グラフィック，テキスト，スプライトの表示優先順位を決定します。図1でPR0～PR3までの部分にはグラフィックプレーンの優先順位が入ります。もっとも優先されるのがPR0です。それぞれ2ビットでプレーン番号を指定します。

Gpr，Tpr，Sprにはグラフィック，テキスト，スプライトの優先順位が入ります。

00＝もっとも優先される
01＝2番目に優先される
10＝いちばん下になる

のようなビット構成をそれぞれに書き込みます。

レジスタ3は表示制御に関するレジスタで，ポートは$E82600_H$から2バイトです。

GS0～GS3は512ドットモード時の表示制御で，GS0がもっとも優先度の高いグラフィックプレーンを，GS3がもっとも優先度の低いグラフィックプレーンを表します。それぞれに，

0＝表示しない
1＝表示する

を書き込みます。

同様に，GS4は1024×1024ドットモードのときのグラフィックの表示状態を，TON, SONはテキスト，スプライトの表示状態を表します。

このあたりがよくわからなければ，ポートを読み出して現在のビットの設定状態を見てみるとよいでしょう。

そして，レジスタ3の上位バイトが本命の特殊制御関係のレジスタです。それぞれのビット指定は以下のとおりです。

GT 0＝通常モード
　1＝グラフィック，テキストの半透明
GG 0＝通常モード
　1＝グラフィック2画面間の半透明
BP 0＝RESERVE
　1＝G-RAMの内容で領域指定
HP 0＝特殊プライオリティ
　1＝半透明モード
EO 0＝通常モード
　1＝半透明，特殊プライオリティ使用
VH 0＝通常モード
　1＝グラフィックとビデオの半透明
AH 0＝通常モード
　1＝グラフィックとテキストパレット0番との半透明
Ys 0＝CMPCUT(インポーズ時に使用)
　1＝TVCUT(インポーズ時に使用)

＊　＊　＊

いずれの場合も半透明とは「グラフィック画面となにか」の合成を意味します。

では4種の半透明モードについて解説します。

1)　グラフィックとテキストパレット0番

グラフィック画面全体に，テキストパレットの0番に設定された色を合成します。BP, AHを1にしてください。

R3High＝44_H

などを設定します。

2)　グラフィックとテキスト＆スプライト

グラフィックプレーンの最下位ビットが立っているエリアにテキストとスプライトを合成します。

合成に使用するグラフィックプレーンはもっとも高いプライオリティにある必要があります。いちばん下のビットは領域指定に使われるのでグラフィックの色数は普段の半分になります。GT, BP, HP, EOを1にしてください。

R3High＝$1D_H$

などを設定します。

3)　グラフィック画面間

グラフィック画面のうちもっとも上にあるものと2番目のものを合成します。領域指定はいちばん上のプレーンの最下位ビットで行います。やはり色数は半分になります。GG, BP, HP, EOを1にしてください。

R3High＝$1E_H$

などを設定します。

4)　2)＋3)

GG, BP, HP, EOを1にしてください。両方の機能が働きます。

R3High＝$1F_H$

などを設定します。

＊　＊　＊

なお，こういった類の作業をするときに「InsideX68000」は必携ですので，ぜひ手に入れておいてください。

C言語でも特別な関数は用意されていませんし，要はポートのアドレスと設定する値だけの問題ですので，ここでは私の使いやすい環境で答えさせていただきます。要点は見ればすぐにわかるはずですから。

ということで，サンプルプログラムはなんとSX-BASIC上で動作します。どのような方法でも構いませんから，あらかじめグラフィック画面にPIC画像を表示しておいてください。リスト1を実行して全角キーを入れると画面上に半透明の窓を開きます(ウィンドウではありません)。

これで壁紙動画をやると……はい，とっても作業しづらいですね。でもなにかグラフィックを置いといて，パターンエディタを立ち上げて1倍モードでなぞるとかすると結構使えます。なんか質問内容とはかけ離れてしまいましたが，このように半透明モードはちゃんと実用的なことにも使えるわけですね。　　　　(中野　修一)

表1　ビデオコントローラのレジスタ

レジスタ2

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$E82500_H$	╳		Spr		Tpr		Gpr		PR3		PR2		PR1		PR0	

レジスタ3

$E82600_H$	Ys	AH	VH	EO	HP	BP	GG	GT	╳	SON	TON	GS4	GS3	GS2	GS1	GS0

リスト1

```
int i,j
int x0=0,x1=511
int y0=0,y1=511
for i=x0 to x1
  di()     :/* ひたすら待つ
  for j=y0 to y1
    pset(i,j,point(i,j)+1)
  next
  ei()
  print i
next
repeat
  if (shiftkeybit and 16384)=16384 then {
    pokew(&he82500,9444)
    pokew(&he82600,63+29*256)
  } else {
    pokew(&he82500,1764)
    pokew(&he82600,63)
  }
until 0
end
func point(x,y)
int adr=&hc00000
adr=adr+x*2+y*1024
return(peekw(adr))
endfunc
func pset(x,y,c)
int adr=&hc00000
adr=adr+x*2+y*1024
pokew(adr,c)
endfunc
```

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので電話番号も明記してください。

宛先：〒103　東京都中央区日本橋浜町3-42-3
ソフトバンク株式会社出版部
Oh!X編集部「Oh!X質問箱」係

FROM READERS TO THE EDITOR

新しい年の始まりです。去年あった嫌なことはすっぱり忘れてしまいましょう。まずは，暖をとってゆっくり休んで考えますか。考えることが性に合わないという人は，最初から飛ばして息切れしないように注意してくださいね。

◆私は嬉しい。やっと，やっとX68000XVIを買ったぞ～。えっ，いまはX68030の時代だって……。しかし，6年間も初代を使っていた私にはすごい速さだ！　やっぱりX68000よね。しかし110,000円もするとは……。

橋本　学(21)三重県

いまやX68000XVIはX68030よりも手に入れにくいかもしれません。それを思えばいい買い物かも。デザインも悪くないしね。

◆森山氏の恐竜には，驚きました。写真だけでもすごいと思っていたのに，Oh!Xを読み終わって「さー今度はMacUserを見よ。おっとその前に，CD-ROMのMacBinでも見るか」ってMacBinを起動したら，なんと森山氏の恐竜が出てきたではあーりませんか。思わず「ゲゲなんだ，これほんとにCGAシステムで作ったんか」とショックを受けました。　山本　勝(22)神奈川県

本当にCGAシステムで作られています。皆さんも負けぬよう精進しましょうね。

◆昔，20MバイトのHDを所有していたとき，6つの領域に分けて使用していた。う～ん。そう，あの頃は若かった。19歳から20歳(ハタチ)になったそのわずかの間に，自分がすごく年を取ってしまったように感じる。皆さんはどうなんでしょう。　角谷　光憲(20)愛知県

昔は20Mバイトでもずいぶん広かったんですけどねぇ。とはいえ，6分割とは……。

◆ついに「H.A.R.P」が届きましたか。最初の予告から1年(！)ですね。Oh!Xらしい辛口のレビューを期待しています(私は半年で堪えられなくなって返金してもらったが……)。

金子　聡史(20)千葉県

もう1年がたってしまったんですね。先月号でレビューをしましたが，期待どおりだったでしょうか？

◆釧路で唯一X68000のソフトを置いていた店から，とうとうX68000用のソフトがなくなってしまいました。理由は「売れない」のひと言だそうです。私ひとりが地道に買っていてもダメだった。悔しかったので，持っていないX68000用のソフト10本を20,000円に値切って買ってきました。　石黒　明彦(24)北海道

10本20,000円というのはお買い得ですが，これから手に入らないとなるとちょっと問題ですね。

◆急激に涼しくなって，うちのX68000も誤動作がなくなり元気いっぱいです。

斉藤　大(20)埼玉県

人間のほうがダウンしませんように。

◆須藤さん，クラウザーは「ヨーグルトいっぱい食べーる」っていってないように思えるんですけど。自分は昔から「悟空，一発だ」というふうに聞こえていて，クラウザーは悟空を挑発しているのだと思いました。久しぶりに「餓狼SP」を立ち上げて聞いてみたけど，やっぱり「悟空，一発だ」でした。人それぞれなのね。

金井　宏明(19)千葉県

でも，悟空なんか「餓狼SP」に登場していないから，挑発してどうするんでしょう。

◆とうとうだまされて「SX-WINDOW ver.3.1」を手に入れた。と，同時に1,000円で「ロボットコンストラクションR.C.」を手に入れた。さらに「スーパーストII」を手に入れた。そんなとき，Oh!XでTeX入門講座を始めたので，TeXもほしくなった。モデムも3,000円で手に入りそうで，いったいどれをすればいいのだろう。

吉留　善行(20)佐賀県

時間の許す限りすべてのものを使いつくしてくださいね。

◆昔，セキセイインコを飼っていました。その鳥は「春はあけぼの」をすべて覚えて，しゃべることができたのです。しかし，七夕の朝，息を引き取り……う一悲しい。そこで，オカメインコのひな(オスかメスかわからなかった)を飼うことにしました。ところが，この鳥，自分の名前もろくにしゃべれない。なぜだろうか？

石井　義尚(19)神奈川県

自分につけられた名前が気にいらないのかもしれませんね。別の名前を考えてあげてはどうでしょうか？

◆うお～。9月の台風ラッシュで雷ピカピカ，停電ばっか。雷のせいでモデムが死んだー。出費の秋よ。早く去ってくれー。

石井　貴光(23)奈良県

秋も過ぎ，いまは冬の真っ最中です。今度は暖房で出費がかさむかな？

◆どんどん寒くなりますが，暖房手段の乏しい僕の部屋では，X68000が暖房器具として活躍してくれるでしょう。　加藤　安弘(20)滋賀県

やはり，本体は足もとに置いて頭寒足熱にするんですか？

◆「スパII」のゲームレビューで「9.95！　新体操～」とありましたが，新体操では逆立ちすると減点されます。逆立ちするのは，女子体操のほうです。──前体操部部長より

杉浦　竜夫(18)岡山県

全然知りませんでした。でも，女子体操も普通は2人で演技をしないから，シンクロナイズドスイミング？　水中じゃないって。まあ，それは風呂場ということで……。

◆会社の行事で，自動車の運転研修に行きました。なかでも怖かったというか，おもしろかったのはスキッドカー。いわゆるタイヤがすべる状態で運転するやつ。8人中7人がポールにぶつかりました。　谷川　正洋(23)広島県

使う車が自分の車だとしたら……。

◆我が家の末妹のパー子(X68000EXPERT)が永

◀吉田　淳一　宮城県
新年のお便りをありがとうございます。でも，受け取ったときは11月の上旬。2月号が希望ということでしたが，やっぱり1月号ですよ。

◀岡村　直也　兵庫県
電話をしながらほかのことをしていると，ついつい相手の話を聞き損ねてしまいません？　皆さんもそんなことをして相手にふられませんように。

眠しました(合掌)。そんなわけで，NEO・GEOで遊んでいます。京や紅丸もいいけど，違いのわかる女子高生はやっぱり，チョイ＝ボンゲですっっっ(「餓狼SP」以来，すっかりゲテモノ好きになりました)。　綿引　一代(15)茨城県

新たな弟妹を購入されることをお勧めします。手術(修理)による復活も可。

◆最近になって美神令子のヌイグルミ(UFOキャッチャーのやつ)の下着が着脱可能であることを知った(笑)。誤解を招くと嫌なので一応書くが，私は3回しか試していない(おい)……。私はもしかしたら(で)氏よりアヤしい(失礼。だって本当にアヤしいんだもの)奴かもしれない。

平野　鉄之助(18)長野県

世間は広く，誤解される人もたくさんいますが，つき合ってみるとほとんどまともな人なんですけどね。あの瀧氏だって……。

◆今年は杉の花粉がとても多いそうだ。いつもなら3月頃に症状が出るのだが，なんだか現在の段階でも目が少しかゆい。大学入試の日が地獄にならなければいいが……。

黒澤　典義(17)群馬県

杉の花粉は春先に出てくるものだから，いまかゆいってことは半年以上も空中をさまよっていたのでしょうか？

◆11月号の「美容院さりん」を私も発見。「SARiN」と看板にあるので，「きりん」の棒が1本ない，というのではありません。音の感じがかわいらしいのでつけた名前だとは思いますが，店の人はどう思ってらっしゃるのでしょうか。

安井　百合江(20)愛知県

名前が変わっていないところをみると，案外宣伝になったと喜んでいるかも……。そんなことはないですかねぇ。

◆ホビーユーザーが多いといわれるX68000ですが，理容業界(床屋)や美容業界には，根強く残っています。うちの店も導入しています。ソフトを作っている会社が愛知県なので，愛知を中心に多くの美容院や床屋でX68000の姿を見ることができます！　皆さんも探してみてください。　市川　博基(19)愛知県

あの「美容院さりん」も愛知県だったから，使われているかもしれませんね。

◆11月号の汗の塩分の話ですけど，確か汗の塩分で塩ラーメンを作るというのが，かなり前にテレビで放映されたような記憶がある。内容は北海道(？)の某高校の生徒たちが自分たちの汗を集め，そこから取り出した塩を使って塩ラーメンを作り，校長に食べさせるとかいうものだったと思うが……。　本多　和正(19)福岡県

そんなラーメンを食べさせられた校長先生と生徒の関係って……。まさか，兄貴な関係(？)。

◆先年，不況のあおりを食らって，勤めていた会社がツブレて以来，仙人のような生活を送っています。もはや，1万円札の顔は誰なのかさえわからなくなってしまいました。が，しかし，Oh!Xだけは欠かさず(姉に泣きついて)買ってきて，パソコンライフを充実させています。

▲鈴木　貴久　神奈川県
どのキャラクターも若々しくていいですね。特にマニュアルのイラストを見たあとだとそう感じます。でも、テリーがちょっとガキみたいかも……。

▲森本　真　愛知県
「スナッチャー」はいい出来ですが、X68000では出てなくて残念です。ところで「E」はもとからこの位置ではないですよね？(ちょっと弱気)

田中　健治(21)岡山県

親切なお姉さんに感謝の毎日ですね。

◆いままでの環境(X68000PRO，メモリ2Mバイト，HDなし)にたえられず，大阪までHDを買いに行ってきました。まず驚いたことは，モノが安い。愛媛ではパソコン本体をはじめ周辺機器にいたるまで，安くて3割引きなのに大阪では半額に近いモノばかり。思わず2Mバイト増設RAMボードとソフト8本も買ってしまった。次に驚いたことは，自分の金遣いのアラさ。愛媛に帰れなくなってしまい京都の姉の家へ金策に……。　山口　大賀(21)愛媛県

クレジットカードは持っていないようですが，手に入れるのはやめたほうがいいかもしれません。

◆1年半前のことだった。教習所で私たちは「スピードコンペ」というものを行いました。それは無線教習でトップスピードを競うものです。結果は私が1位(70km/h)でした。コツは4速に入れないことです。皆さんがんばってください。

西浦　宏和(20)愛知県

残念ながら，無線教習を受けたことがないので状況がよくわかりませんが，あまり無茶をしませんように。

◆雫石駅にある自動販売機に，「暖かいジュースあります」と書かれた張り紙があった。この文がおかしなことに私はすぐに気づいた。まず，「暖かい」がおかしい。「温かい」が正しいのだ。次に「ジュース」がおかしい。「ジュース」を旺文社国語辞典第八版で調べてみると，「果実，野菜をしぼった汁」とある。だが，その自動販売機にあるオレンジジュースは温かくないのである(紅茶やコーヒーはHOTなのだが)。てなわけで正解は「温かいコーヒー，紅茶，ココア，お茶あります」だと思うのだが，くどいので「温かいお飲物あります」。これでキマリだ！

大久保　明弘(22)岩手県

いきなり国語辞典を引っ張り出すとは，想像するとちょっと怖いかも……。

◆近所のパチンコ屋に大きな文字で「やったぜ！　終了!!」と書いてあるのは，つまりその店自体が終了してしまったことを指しているのだろうか？　堂領　輝昌(20)神奈川県

そこは，一度入って終了を味わってみるのがいちばんかもしれません。でも，持っていったお金が終了しても当方は関知いたしません。

◆今年，就職したのですが自分に合わないとわかったので辞めようかと思っています。東京は住むのに向かないといま感じています。人生には車と68030が必要ですネ……。わずか6カ月で辞めるのは早いでしょうか？

高橋　秀一(21)東京都

自分でそう決めたのなら，それは仕方のないことです。後悔のないようがんばってくださいね。ただ，東京でも車と68030は持てますよ。

◆とある友人(♀)に聞いたのですが，女性の服の流行というのは「流行を決める団体」が，2年前にその年の流行を決めてしまうのだそうです。つまり，1994年の花柄とシースルーは，1992年にどっかの誰かさんたちが作った流行なわけです(実話)。女性の方々，踊らされないように注意しましょう。ちなみに来年の流行は……。

片倉　純也(20)宮城県

流行なんて気にしている人が，どのくらいOh!Xを読んでいるんでしょうか？　だって，そんなことをいったらX68000は……。

◆自炊し始めて2年半，ついにフライパンから炎をあげることができた。鉄人となる日は近い。

吉田　務(22)大阪府

次は鉄人に挑戦ですか？

◆11月号の岡村さんの4コマではありませんが，最近，AD PCMを使って似たようなことをしてます(某声優さんの声を集めてたりする)。

鈴木　俊之(19)静岡県

あまり危険な声を集めて，世間の人の誤解を招きませんように。

◆先日，情報処理試験監督員のバイトをしたのですが，よくもまあ，あんなのものを1時間でできるものだと感心しました。ただ，選択問題のマーク記入に注意することと，問題用紙の表紙をよーく読むことをこれから受験する人にいいたい。　原田　健史(27)東京都

記入を間違ったり，忘れた人を見かけたんですね。確かにこんなことで落ちたらもっ

たいないですからねぇ。
◆がんばれ，Oh!X！　正しいものが負けるものか。　三浦　栄悦(26)秋田県
歴史を振り返ると……。
◆部屋の大移動を行い，念願の１人部屋となりました。ついでにX68000も持ってきて「これで夜中でもパソコンに触れる！」と喜んでいたのもつかの間。ファンの音がものすごく気になるんです。「徹夜でプログラミング」の夢は遠そうです。しかし，なんであんなにうるさいんだろう。　澤田　恭幸(17)千葉県
な〜に，それに負けないくらいの音を周りで鳴らしていればいいじゃないですか。そのなかでプログラミングに集中できるかはあなた次第ですが。
◆コナミさん，「スナッチャー」まだですかぁ〜？　来月には発表かなぁ〜。「ラムちゃん，海岸にあるのは？」「スナッチャー(砂っちゃー)」……。　尾形　敦(19)東京都
……。コナミさん，怒らないでくださいね。
◆ようやく，ハードディスクを購入することができました。SCSIボードとしてX68000CompactXVIを買いました。結局，某番組の陰謀(６月号155ページ)にまんまとはめられたような結果になってしまったわけです。増田　和通(22)東京都
街を歩いているときにキャッチセールスにつかまらないよう気をつけてくださいね。
◆仕事でパソコンを使う女性に不妊症が多くなっているらしい。男の場合でもなにか不都合が起こるかもしれない。PCM，CGもよいがやはり人間には自然の音，風景が必要なのだと感じた。いまではバーチャルリアリティまで体験できるが，ふだん何気なく生活している空間こそ最高のバーチャルリアリティなのかもしれない。
相方　道宏(25)広島県
それが現実であろうと仮想現実であろうと，本人が現実だと感じるのなら問題ないと思うのですが……。
◆先日，歯医者に行って歯の神経を取りました。そのとききちんと麻酔が効いてなくて，ものすごく痛かったです。それからもその歯医者に通っていますが(ほかにも虫歯があった)なんか不安です。　三津田　哲雄(19)山口県
歯は抜いてしまうと，あとがないですから，安心できる医者がいいと思いますよ。
◆秋休みは，深夜バイトで昼夜が逆転してしまった。後期が始まって……，あー眠い。
井南　尚幸(19)山形県
後ろから教授の影が忍び寄って……。
◆パソコンにとって快適な季節がやってきました。うちのX68000XVIは，現在24MHzにしてもビシバシ動いてくれます(夏はアドレスエラーがビシバシだった……)。ところで，X68030が品薄だと聞いたのですが，金が貯まる頃には買えるでしょうか(不安)。　永田　武士(20)東京都
人間にとって快適な季節とはいいがたいのですが……。
◆２，３日前，私はかの名機ローランド「U-220」の99%新品を見つけてしまった。箱などはすべてあり。店の展示用でデモなどの音源としての使用はいっさいしていない。外側が人の手アカと空気で汚れただけ。そして，どうしてもこれを手に入れようと，恐る恐る価格を聞いてみると，な，なんと29,800円でいいというのだ。もちろん，いまは家のスピーカーからナイスなサウンドが鳴っています。
池田谷　公一(19)大阪府
まずは，聞いてみるもんですね。ほかの人もお店のウィンドウをしっかり眺めてみたら，掘り出し物があるかも。
◆ウチの車のファミリア君も７歳になる。しかし，右前足の調子が悪い。ダンパーがだめなようである。乗り潰すつもりで……と考えていたが，修理見積もりを見てびっくり。「買い替えどきでは……」といわれて，「まだ……」と思ったがFTO(三菱の新車)が気になる。
谷川　正洋(23)広島県
乗る時間が長くなるほど愛着も湧くものですね。でも，命にも関わることですし，仲よく右足を痛めたりしませんように。
◆とても寒いです。まだ10月なのに家の中でセーターを着ています。ときどき仕事に出かけるときに，も〜っと寒い外に出なければならないのが，すんごくしんどいのです。でも，TAKERUに寄るときは元気なのでした。主婦は風邪をひいても休めないのがつらい。熱があっても洗濯，掃除，買い物，ダンナの相手，咳が止まらなくても，晩飯の準備。……晩飯は大好きです。
三木　陽出(34)北海道
好きなことをしていると少しくらいつらいこともがまんできるものです。ということで，これからのお出かけは帰りにTAKERUに寄るというのはいかがですか？　え，そんなにタイトルが……。
◆CGA大学，早くも留年確定か？　というぐらいややこしいですね。なにしろ頭が弱いので……。　津村　忠蔵(19)佐賀県
なあに，いくら留年しても自分があきらめないかぎり退学はありませんので，地道にがんばってみてくださいね。
◆11月号の井上さん，私も174cm，45kgの超ヤセ型です。寒さに弱いのをはじめとして，プールで浮かない，自動ドアが開かない，強風で飛ばされるなどの苦労の数々，なかなか人にはわかってもらえません。　松永　貴輝(24)大阪府
自動ドアが開かないなんて……。もしかして，空中浮遊の術でも心得ておられるのでは……。
◆全身麻酔ってすごいです。もやもやしたものが意識の中に出てきたとたん，無限の闇に突き落とされます。慣れてくると快感という話も。
岡島　隆史(25)奈良県
このタイプは一気に意識を失うタイプですかねぇ。ほかにも，温度を感じなくなって，触覚や聴覚が失われていくものもあるそうなので，一度試して……みないほうがいいような。
◆立ち読みのためアスキーを取る，Mac Userを取る，ベーマガを取る。買うためにOh!Xを取る。……Oh!Xってこんなに軽かったか!?　がんばれOh!X。　中嶋　祥史(27)神奈川県
これも日頃のダイエットのおかげ(悲)。
◆北海道から静岡県へ転勤して，はや半年。妻へのご機嫌とりから，県内，県外の観光地巡りに毎週末追われて，X68000君に触れるのは週に１〜２時間，ゲームをするときだけです。11月号の特集を読み，またBASICに挑戦意欲が出てきました(先月は上高地，清里，北軽井沢，伊豆へと遊びに行き，もうヘトヘト)。
坂本　慎太郎(34)静岡県
たまにはX68000君にもご機嫌とりをよろしくお願いします。
◆会社の寮を出てひとり暮らしを始めました。半年ぶりにX68000XVIに触れております。でも，ひとり暮らしということは，これでいよいよ念願の……(以下略)。　上田　考一(24)広島県
省略なんてしないで，教えてくださいよ。あんなことやそんなことや，なんか怪しげなことを考えていますね。
◆関西へ修学旅行に行ってきました。それはよかったんですが，日本橋でSX-WINDOWを買ってしまい，おかげであの箱を持って札幌に帰るハメに……。関係ありませんが，丸１日列車に乗っていると，降りてもあと１日は列車に乗っている感じが味わえます(あ，地面が揺れる)。

◀板橋　芳則　福島県
このタイトルは「コンピュータの女神」。そこで，この女神の名前を募集するそうです。お気に入りの方は連絡をお待ちしています。
◀岩瀬　貴代美　福岡県
この美しい女性は自画像ですね(しっかり11月号のフォロー)。そうじゃなくて，イレーネ嬢と書いてありました。また間違えてしまうところでした。

平間　大輔(17)北海道

船に丸1日乗っているというのも結構きますよ。飛行機でエアポケットにハマるのもジェットコースターみたいでいいけど。

◆天皇賞を当てて，ハードディスクを買おうと思った。……外れた。今度は菊花賞だ。絶対当てて買うと友人にいったら，まじめに貯金したほうが1年早く買えるぞといわれた。

渡　貴弘(21)大阪府

1年ではすまないかもしれません。でも，逆に1年でMIDI音源やモデムがついてくることも……。

◆ハロウィンでアメリカのどこかの放送局がやった“地球にたくさんの隕石が落ちてくるぞ”というニセのニュース，日本でやったらジョークじゃすまないよね。でも，1回くらい日本でやってくれないかな。　海川　文彰(25)長野県

エイプリルフールの頃に雑誌なんかではたまに見かけますけど。そうすると，東スポは……。

◆「あすか120%の秘密」(1)一めぐみのボンボン一空中でめぐみがボンボンアタックをすると反動でめぐみの体が少し押し戻される。このことを考慮すると，ボンボンの重さは2つで14kg前後あるものとみられる。あなおそろし。

島田　貴之(20)埼玉県

見かけによらず力持ちなんですね。そうじゃないって。

◆1%を切る女子高生のうちの1人です。5カ月後，もっと少ないであろう「女子浪人生」になってたりして。きゃはははは……はぁ(ため息)。

松尾　絢(18)長崎県

どんな読者であろうと大歓迎ですよ。あ，フォローになってないですね。

◆お煎餅の町，草加の隣に住む私は，「湿った」煎餅がなかなかイケることに気づきました。「しけた」ではありません。しょうゆのしっかりしみた焼きたての煎餅です。堅すぎず，柔らかすぎず表面の湿った感じがとってもジューシー。

中島　民哉(24)埼玉県

とっても，美味しそうですね。でも，これって煎餅屋に実際に行かないと食べられないから……。仕事を抜けて煎餅屋を探したくなってしまいました。

◆先日，某大学の学園祭に行ったときの話。建物の中のある部屋で，X68000SUPERで遊んでいる人を発見。しばらく話をしていると，その大学の学生が，後輩が彼と知り合いだったらしく，彼に話しかけた。「あ，小川君。元気？」。その瞬間，私の頭の中にOh!X誌上で見慣れたイラスト常連投稿者の名前が浮かんだ。「も，もしかしてあの小川伸輔君？」「そうです」。うわぁ(笑)，世間は狭いっていう話は本当だったんだ。ああ，びっくりした(そのあと，すかさずサインしてもらったミーハーな私)。遅ればせながら合格おめでとう(1994年5月号参照)。またSUPERちゃんの4コマ描いてね。　高橋　洋一(21)千葉県

最近イラストを見ないのでてっきり封印されているのかと思っていました。小川さん，また，イラストを送ってくださいね。楽しみにしています。

◆「餓狼SP」のストーリーモードでリョウ・サカザキで遊ぶ方法を教えてください。

佐藤　光一(28)宮城県

知ってる人がいたら，よろしくね。

▲奈良原　仲哉　福岡県

コンピュータに接し始めて最初のプログラミング言語が機械語とはなかなかドラゴンですね。ところで，プニグラマってどういう意味ですか。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★サークル「X'm」では活動再開にあたり，新規会員を募集します。対応機種はX68000で，月刊のディスクマガジン発行が主な活動になっています。興味をもたれた方はフロッピーディスクを2枚と返信用の切手を同封して下記の住所へ送ってください。〒604　京都府京都市中京区釜座通り夷川下る大黒町686 森薫様方　5号室 柴田　泰史

売ります

★48ドット熱転写カラープリンタ「CZ-8PC5」をリボン数個(黒，カラー)ずつつけて10,000円で売ります。送料は込みです。また，説明書はありますが，箱はありません。連絡は往復ハガキでお願いします。〒825　福岡県田川市大字伊田4271 メゾンドホリ112　矢上　裕之(19)

★X68000CompactXVI用2Mバイト増設RAMボード「CZ-6BE2D」を18,000円で売ります。送料は込みです。連絡は往復ハガキでお願いします。〒514-01　三重県津市栗真町屋町1052-1　ハイツ坂野105号　兼松　隆文(24)

★カラーイメージスキャナ「JX-220X」を60,000円で売ります。付属品などはすべてあります。新品同様です。連絡は往復ハガキでお願いします。〒079-12　北海道赤平市平岸新光町5-2　北城　彰也(19)

★ローランドの音源モジュール「SC-55」とシステムサコムのMIDIボード「SX-68M」をセットで35,000〜40,000円で売ります。どちらも箱，説明書はあります。連絡は往復ハガキでお願いします。〒986-03　宮城県桃生郡桃生町樫崎字高附137-2　佐藤　泰満(35)

★ADTECのDOS/V用72ピン8MバイトSIMM(60ns)1枚を20,000円以上で売ります。また，X68000XVI用2Mバイト増設RAMボード「CZ-6BE2A」を20,000円以上で売ります。連絡は往復ハガキでお願いします。〒674　兵庫県明石市大久保町大窪1352-11　佐藤　浩司(22)

買います

★MIDIボード「CZ-6BM1」かシステムサコムのMIDIボード「SX-68MII」と音源モジュール「SC-55」以上のものをセットにして33,000円で買います。また，エヌエス・カルコンプのタブレット「Drawing Slate(31090SER)」を35,000円で買います。完動品であれば，箱や説明書がなくてもかまいません。連絡は官製ハガキでお願いします。〒811-01　福岡県粕屋郡新宮町下ノ府1070-28　佐藤　光司(24)

★SCSIボード「CZ-6BS1」かシステムサコムのSCSIボード「SX-68SC」を13,000円で買います。連絡は往復ハガキでお願いします。〒860　熊本県熊本市島崎5-45-20　竹下　雄一郎(17)

★SCSIボード「CZ-6BS1」を12,000円で買います。付属品と説明書がついているものをお願いします。連絡は官製ハガキでお願いいたします。〒213　神奈川県川崎市高津区上作延903　間　晴樹(24)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は11月号の内容に関するレポートです。

●私はどうもX-BASICが好きになれないので，SX-BASICの文法をCにしてくれる人はいないのでしょうか。BASICでもCでもどちらでも動くようにしてくれるともっと嬉しいですね。リストの先頭で〝Hello Hello World〟と書いておくとCモードになるとかね。結局，11月号の特集は，私はあまり興味がありませんでした（やっぱりCが好きなもんで）。しかし，Oh!Xの誌面を賑やかにしていこう，というような感じでSX-BASIC，TGRAPH.FNC，XSPRITE.FNCはよかったと思います。

あと，基本的に私はCの人なので，「Cを知りたい」と問われれば，どのあたりを教えればいいのかだいたい想像がつきます。しかし，「BASICを知りたい」という質問はあまりされたくありませんし，答えられないでしょう。というわけで11月号の特集は，BASIC入門として「良」かどうかはよくわかりません。文法の説明は中野氏の記事のみだけど，その内容は「慣れろ」ということです。確かにBASICがわからない人は「プログラミングがわからない」というレベルにないので，そう考えると入門として今回の特集は悪くないのでしょう。しかし，11月号の特集で「なるほど！BASICとはこういうものか」わかった人は，いないと思えました。

中村　俊之(20)　X68000 EXPERT,X1turbo model30,MZ-1500/700,PC-286noteF,PC-E500　東京都

●手軽にX68000の機能に触れるには，またプログラミングの入門には，X-BASICは手頃な言語でしょう。プログラミング入門としてこの言語を特集したことは決して間違いではないと思います。しかし，全体的に内容が難しく感じられ，入門というよりは中級者向けではないでしょうか。それに，今回の内容で，初めてプログラミングしようとした人でも理解できたかどうか疑問に思います。

また，BASIC入門としてよかった点は，特集の最初の記事が初めてX-BASICに触れる人にもわかるような内容であったことです。逆に悪かった点は，初心者には難しいと思われる記事が多かったことです。これについては，少しプログラミングができる人（少しは経験がある人）にとっては，よかった点に変わるでしょう。

壁谷　善嗣(35)　X68000 EXPERT,PC-9821As/9801NS/E　宮城県

●11月号の特集は，年に1回あると思われるBASICについてですね。X-BASICは，X68000に標準で付属してくるもっとも身近なプログラミング言語。そのX-BASICについての特集ですが，今回はかなりわかりやすい記事でしたね。これならプログラムを作りたいけどマニュアルだけでは……といった人たちにも理解できたかもしれません。しかも，SX-BASIC入門，外部関数の作り方などもあって，非常に内容の濃いものだったと思います。

特集の中では，「BASIC修得への道」のコラムや，「BASICからの離脱」などがよかったですね。私も自分のプログラムがとろいのか非常に重たいものばかりで，高速化のための技はいろいろ使ってきました。それにしても変わった乗算の方法があるのですね。さらにBAStoCを使ってGCCも使用するなんて思いつきませんでした。さっそく実践してみます。

大上　幸宏(21)　X68000 PROII　鹿児島県

●新製品紹介にあった「BJC-400J」。いやはやなかなかきれいではないですか。これで定価が69,800円とは。「MJ-700V2C」といいNECの新しいプリンタといい，そろそろカラーインクジェットプリンタも一般的になったということでしょうか。

あとヤマハのXGは，なかなかすごい企画ですね。かなり興味はあるのですが，音源というのは1台1台の特徴があってこそ面白いものです。その音源の限界まで性能を引き出そうとすると，統一規格のMIDIデータなど意味がなくなってしまうし，なかなかXGの存在意義も難しいところですね。しかし，1台欲しくなる性能と価格なのは確かですね。

そして，松下のPDはなかなかおいしい機械なのですが，普及するかどうかがポイントでしょうか。CD-ROMやMOの普及もかなり進んでいることですし，どうなるか興味があるところです。X68000にもSCSIを用いればなんとかなりそうですし。

ハードディスクの代わりにはならないでしょうが，CD-ROMが読め，データディスク，ハードディスクバックアップ用のメディアとして650Mバイトのディスクが使える。こう考えるとなかなか使えるドライブかもしれません。しかし，テープにデータをセーブしていた頃を思うとすごい進歩ですね。

奥田　直也(22)　X68030/X68000 ACE-HD/SUPER，MSX2，PC-E550　愛知県

●僕自身，BASICでゲームを作ろうとして，マニュアルとにらめっこしたことがありますが，グラフィックを扱おうとした段階で突然難しくなったと思いました。11月号の特集では，グラフィックに関する基礎の基礎をもっと詳しく解説してほしかったですね。

MDデータが発売されるようですが，どうも失敗する要素が多いような気がしました。音楽用MDは使えない，録音ができない，安いのはいいけどMOが普及しすぎている，遅いなど。あと少し，という要素が多いのが残念です。

野田　類邦(20)　X68000 EXPERT,PC-286VG　愛知県

ごめんなさいのコーナー

12月号　ペンギン情報コーナー

P.138　ビーピーエス社から発売になった音響カプラの型番に誤りがありました。正しい型番は「XAC-1」です。ご迷惑をおかけした関係者各位にお詫び致します。

12月号　垂直帰線期間待ち外部関数

P.61　外部関数vdisp()でレジスタ保存を行っていないため，インタプリタ上で正しく動作しても，コンパイルした場合に不都合が生じるバグが見つかりました。

リストの32行と33行の間に，

```
movem.l d0-d7/a0-a7,-(SP)
```

47行と48行の間に，

```
movem.l (SP)+,d0-d7/a0-a7
```

の2行を追加してください。

バグに関するお問い合わせは
☎03(5642)8182(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

割り切ればそこにはパラダイスが……

▶今月は「割り切って使うCD-ROM」と題して，X68000でCD-ROMをどのように活用すべきか，どのようなものが使えるかレポートしてみました。現在のところメーカーサポートの弱さからデータの利用のみ，と活用法は限定されてしまいますが，巷に溢れるCD-ROMのなかには，ちょっとした工夫によって使えるものもあります。それに，メディアとしての善し悪しはともかく，あくまでも大容量データ供給媒体としてみると，CD-ROMもなかなか捨てたものではありません。

結局，CD-ROM導入のポイントは，大容量のデータを思う存分活用できると考えるか，データプレイヤーだったらいらないと考えるかでしょう。ドライブも安くなりましたし，これから周辺機器を充実させようと計画している人は，選択肢のうちのひとつに加えてみてはいかかでしょうか。

▶石上氏がボヤボヤしている間にリリースされた，X68000XVI用68030アクセラレータ「Xllent30」。「このボードを使えばあなたのX68000 XVIがX68030に！」とまではいきませんが，それに近い性能を発揮できるようになります。定価も59,800円とべらぼうに高いわけでもありません（かといってほいほい手が出るほど安くはありませんけど）。X68030を買うお金はないけど，マシンパワーがもう少し欲しいな，という人は検討してみてはいかがでしょうか。なお，今回の製品はX68000 XVI専用ということでしたが，10MHzのX68000シリーズにも対応する予定もあるようで，ちょっと期待したいところですね。

また，別のところからは，DSPボードなるものの発売が予定されています。こちらも製品が届きしだいレポートしますので，お楽しみに。

▶「ハードコア3DエクスタシーSIDE B」は著者多忙のため，「X68000マシン語プログラミング」は，ドツボにハマってトッピンシャンから脱出ならず，ということでお休みとなってしまいました。ごめんなさい。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたフロッピーディスクを添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒103 東京都中央区日本橋浜町3-42-3
ソフトバンク出版部
Oh!X「㋢㋮㋴」係

SHIFT・BREAK

▶カラーサブノートを衝動買いしてしまった。これからの季節は，やっぱりノートだよな。こたつに入りながら原稿書けるし，ブレーカーが落ちても無停電電源装置がついてるし（バッテリともいう）。あと部屋にLANでも組んだら極楽だなぁ。でもこれでSATURNもPlayStationも買えなくなってしまった。あっ，こたつ布団買わなきゃ。（I.K）

▶MS-DOSのバージョンアップをしたら，問題続発でもう大変。「ハードディスクを認識しない！」と思ったら領域確保を忘れてたり，「FEPを組み込めない，取説にも載ってない！」と思ったら補足取説にデバイスドライバの記述。圧巻はFEPのメディアコンバートができないと思ったら，編集部のブランクディスクが2DDだった。こりゃわからんぜ。（E.K）

▶バイト先で有線が流れっぱなしになってるので流行の歌はほぼ覚えてしまう。そういう歌はたいていイタイのが多いのだが，驚くのはそのサイクルの速さ。1週間で姿を消すものもザラにある。そういう泡のような歌に囲まれながら，漫画にしろ映画にしろ本当に残っていくのはどんなものなんだろうと考えながらレジを打つ今日この頃である。（哲）

▶次世代ゲーム機がどうのこうのと騒いでいるが，ラインナップを見ると極端にシューティングが少ないので論外。それならいっそ基板のシューティングでも買おうかと思ったが，そもそも近頃ロクなシューティングが出ていないので，手頃なものが見当たらず，そっちもダメ。業務用のNEO・GEOでも買ってみようかな。ソフト安いしね，うんうん。（八）

▶12月号のバニーさんシミュレータのメーター部分が，読者の方が投稿したものに酷似していると抗議のハガキをいただきました。作るときにはまったく意図していなかったのですが，投稿チェックで見たものが頭に残っていたのか，結果的に似たような構成になっていたようです。今後このようなことがないよう気をつけます。申し訳ありませんでした。（で）

▶おもちゃ屋の前で大人も子供もバーチャファイター(VF)の精巧な動きに魂を奪われていた。そこがVFの凄さのひとつだし，それを実証したのはセガの功績だ。VF2では酔拳使いの動きが印象的。作者の入れ込みようが伝わってくる。恐るべき仕事ぶりだ。というのもおこがましいが，3D野郎のひとりとして，遅れを取りまくった気分なのだ。（A.T.）

▶11月22日。ついつい買ってしまった「SATURN」。もちろんバーチャファイターが目的。CD-ROMへのアクセスも3DOほど待たされないし，確かにあのバーチャファイターの面白さは完全に表現されている。確かによい。おかげで左手親指も痛い。でもせっかくだから，家庭内プレイヤーならではの構成も欲しかったな，って贅沢か。（K）

▶う〜，寒い。もう初夏だというのに……。これなら日本のほうが暖かい。それに風も強いしな〜。ここはニュージーランド。自然の雄大さに心奪われ，のんびりしたお国柄に日本を忘れる。戻ってきたら，当然のごとく地獄が待っていた（自業自得か）。でも，また行きたいな。今回は同行者の反対にあったが，次こそバンジージャンプだ。（高）

▶年明けとともに「Hello!PC」（はろっぴぃ）編集部に行きます。Oh!Xは本当に読者と共に作っているという感じで楽しかったんだけど，ちょっと魔（？）がさして結婚してしまったのでした（で，9月号のLDの主な使用者は実はあたしです）。Oh!Xでは皆さんと同じ一読者に戻りますが，これからもよろしく。初心者の方は，はろっぴぃも読んでね。（ふ）

▶11月22日，久しぶりにワクワクしながら買い物をした。なんともいえない期待感と，もしかしてなかったらどうしようという不安感が入り交じり，自然と足早にショップを駆け巡る。おかげで足が痛い。すぐさまサルのように遊んで親指がつりそうにもなるし。今度は12月3日に，また秋葉原でもワクワクしながらうろつき回るとしますか。（J）

▶「あの人はまだ手を使ってカーナビを操作している」よそ見運転がいちばん危ないんだよ。アメリカなら「運転中にカーナビを見ていて電柱にぶつかった」とかいってメーカー責任訴訟で儲けられるんだが。んで，自動車電話をかけながら運転しているドライバーを見るとつい飛び出したくなる……危ないから小さな子は真似しないように。（U）

▶がんばったときに拍手をおくるのは当然だが，出来が悪ければ徹底したブーイングをあびせる。もちろん，どんなに批判しても，X68000ユーザーは自分のマシンを誇りに思っている。CPUで負けても，アプリで負けても，周辺機器で負けても（まるで負けてばかりみたいだ），パソコンユーザーとしてのプライドは譲れないから。（ケンゾーやめろ！ T）

microOdyssey

「付録CD-ROM」。まあ，いろんな人から指摘されてはいるが，そろそろ付録ディスクというものも考えなおさねばならない。

では，まず付録CD-ROMが不可能である理由を並べてみよう。

1) ユーザー数が少なすぎる
2) 定価が上がりすぎる
3) 入れるものがない

だいたい上記のようなことになる。

まず，1)だが，現時点でのCD-ROMドライブユーザー数はお話にならないほど少ない。本来なら，せめて過半数がドライブを所有しているくらいの状況でなければこういう話も出てくることがないはずなのだが。

2)の定価の問題。CD-ROMをつけて現在と同じ値段で出せると思っている人はいないだろうが，どれくらい上がるのかなどは見当もつかないだろう。正確な試算は出していないが，だいたいの感じでは特別定価2,500円（税別）くらいが適当か？ といったところだろうか。

現時点で220円アップというものさえなかなか容認してもらえないことがわかっているだけに，これはつらいところだ。人によっては「ディスクなんて1枚50円なのに，どうして？」と，いうことになる。こういう人は原稿料を認めてくれないのだ。

さらにいえば，たまに勘違いしている人もいるが，付録ディスクをつけて定価を上げてもソフトバンクの儲けが多くなるとは限らないのだ（むしろ逆のケースも多い）。

3)は私のまったく個人的な趣味による。要はあまりスカスカのものをつけたくないだけだ。まあ，CD-ROMをつけて内容量が5Mバイト程度っていうのは本誌の読者には納得してもらえそうにないのだが。結局，これは2)とも密接な関係にある。収録するものが多くなればなるほど原稿料はかかるのだ。

次に，あえて付録CD-ROMをつけたほうがよいと思われる点を挙げてみよう。

1) 絶対的なディスク容量不足
とにかく，悩まなくて済む。

2) X68000用としてはもうほかに機会はない
……のかもしれない。しかし，常識的に考えて次期マシンがCD-ROMを標準搭載していない可能性はキーボードやマウスを装備しない可能性と同じくらい低いので（残念ながらそこまで斬新なマシンが作れるとは思っていない），そうなると，いずれは全面的に付録CD-ROMに移行できることになると予想される。とすれば，焦る必要はないか？

3) 集大成がほしい

実際のところ，ユーザー数は少なくても，要するにその1枚でドライブを買っていいと思うくらいの価値があるCD-ROMを作れば問題ないわけだ（と，簡単にいう）。しかし各種開発環境，ツール類，ゲーム，データ類などをかき集めるにしても，Oh!X内部だけでは600Mバイト以上の空間はとうてい埋められない。これには読者の協力が不可欠だが……。

はたして付録という位置付けにするのが正しいのか，別冊形式がよいのか，それも問題だ。

いずれにせよ，そのうちなにか1枚作らねばならないかなと考える今日この頃である。

（U）

1995年2月号1月18日㈬発売

特集 Micro Processing Unit

・各種プロセッサカタログ　・MPU68000命令セット一覧

1994年度GAME OF THE YEARノミネート

いきなり現れた特別企画

次世代ゲーム機を並べてみましょう

新製品紹介　Datacalc SX-68K/AWESOME-X(予定)

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートB1	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	旭屋書店池袋店	03(3986)0311
	八王子	くまざわ書店八王子本店	0426(25)1201
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	043(224)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある『新規』『継続』のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

基本的に，定期購読に関することは販売局で一括して行っています。住所変更など問題が生じた場合は，Oh!X編集部ではなくソフトバンク販売局へお問い合わせください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS（株）にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 1月号
■1995年1月1日発行　定価680円（本体660円）
■発行人　橋本五郎
■編集人　稲葉俊夫
■発売元　ソフトバンク株式会社
■出版事業部　〒103 東京都中央区日本橋浜町3-42-3
Oh!X編集部　☎03(5642)8122
販売局　☎03(5642)8100 FAX 03(5641)3424
広告局　☎03(5642)8111
■印刷　凸版印刷株式会社

満開の電子ちゃん

作・え　岡村（おかむら）祭（まつり）

編集部より

電脳倶楽部も80号を迎えることができました。これもひとえに読者のみなさんのおかげです。

それはそうと、最近68を可愛がってあげてますか？　電クラも2枚組になり、さらに役立つ情報が満載。68用でない周辺機器をつなぐ企画や新製品使用レポートが毎月載ってます。レイトレやプログラミング講座の連載も快調、もちろんツールもキラリと光るものばかり。これを読まなきゃ貴方の68ライフは不幸のずんどこってもんでしょう。

3ケ月4,500円という定期購読もできますので、やめた貴方や読んでない貴方も、ぜひまた宜しくってことでひとつ。

推薦文切れてます。送って下さいね。

THE AKIHABARA EXCITING FESTA

秋葉原電気まつり:11月19日(土)～'95/1月8日(日)

5千円お買い上げ毎に抽選券1枚プレゼント。1等:10万円　2等:5万円　3等:5千円 賞金総額6,000万円

TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKUMO TSUKU

受付時間(平日)AM10:45～PM7:30　木曜定休
(日・祝)AM10:15～PM7:00

『FAX24時間お見積もり受付』
03-3255-4199
お名前.住所.電話番号.FAX番号をご記入の上ご依頼下さい。

ツクモグローバルJCBカード

JCBならではの国内・海外サービスにツクモオリジナルの特典をプラス。ツクモ各店にある入会申込書にてお申し込み下さい。くわしくはグローバル事務局03(3251)9898又は各店へ。
※ジャックス・VISA・セントラル・マスターも取り扱っております。

映像関連機器

動画を始めてみませんか?

ビデオ入力ユニット CZ-6VS1 定価¥178,000

MC68EC020(25MHz)の32BitMPUを搭載し、SCSIインターフェイスを介してパソコンへデータを転送。動画・静止画を簡単に保存出来るアプリケーションソフト「ライブスキャン」を標準装備。1,677万色まで対応し、最大640×480ドットの高解像度で、高速取り込が可能です。但しX680x0シリーズでご使用の場合には6万5千色までの表示となります。

ツクモ特価 ¥142,000

多機能対応型スキャンコンバーター

電波新聞社 XVGA-1v 定価¥66,800

X680x0シリーズやその他のパソコンの水平周波数(24KHz/31KHz)をNTSC標準信号に変換するスキャンコンバータユニットですので、家庭用テレビやビデオ・デッキで映像を表示または録画することができます。また、ビデオプリンターを使えば画面のハードコピーも可能です。

ツクモ特価 ¥56,700

XVGA OVERLAY UNIT

定価¥45,800
「XVGA-1V」に接続して、パソコンとビデオの映像を合成する拡張機器です。

ツクモ特価 ¥38,900

ビデオプリンター(昇華型)

シャープ VP-ES1 NEW

高画質ハガキ大プリント、普通紙・布転写用紙もOK。4分割、16分割、ストロボも可。※入力信号は、ビデオ信号となりますので、パソコンに接続の場合にはお問い合わせ下さい。

ツクモ特価 ¥49,300

コンピュータアートスーパーグラフィックツール

その1 慣れてしまうとマウスがいらない!

DrawingSlate	¥74,800
Matier Ver2.0	¥39,800
合計定価	¥114,600

ツクモ特価 ¥85,000

その2 ハイクオリティなのにこんなに安い!

MJ-700V2C	¥99,800
プリンターケーブル	¥4,800
Matier Ver2.0	¥39,800
合計定価	¥114,400

ツクモ特価 ¥96,000

MIDIコンピュータミュージック特選セット

セット	内容	価格	ツクモ特価
Roland SC-55mkII セット	SC-55mkII	¥69,000	¥79,800
	TS-6GM1	¥39,800	
	MIDI変換ケーブル	¥4,000	
	合計定価	¥112,800	
Roland SC-55mkII 互換セット	MC6600	¥49,800	¥64,800
	TS-6GM1	¥39,800	
	専用MIDIケーブル	¥3,500	
	合計定価	¥93,100	
Roland SC-88 セット	SC-88	¥89,800	¥99,000
	TS-6GM1	¥39,800	
	MIDI変換ケーブル	¥4,000	
	合計定価	¥133,600	

大容量記憶装置

ハードディスク

	ツクモ特価
340MBハードディスク	¥29,800～
500MBハードディスク	¥36,800～

CD-ROMドライブ

メーカー	型番	ツクモ特価
ナカミチ	MBR-7(トレー・7連装)	¥38,800
メルコ	CDS-E(SONYドレー)	¥24,800
Logitec	SCD-400(NEC製4倍速)	¥37,800
エレコム	ECD-250(2倍速)	¥19,800
PIONEER	DRM-602X(6連装2倍速)	¥54,800
AIWA	ACD-300WI	¥23,800
	CD-ROMドライバーソフト+SCSIケーブル	¥9,000

MO特選セット

SCSIボードが必要な場合にはセット価格に¥22,000加算となります。

製品	セット内容	ツクモ特価
Logitec LMO-200 (128MB) ¥79,800	プラス MOメディア SCSIケーブル ターミネータ	¥59,800
Logitec LMO-400 (230MB) ¥158,000	プラス MOメディア SCSIケーブル ターミネータ	¥99,800
ELECOM EMO-L230 (230MB) ¥128,000	プラス MOメディア SCSIケーブル	¥89,800

パソコン通信

モデム

US Robotics Sportster 28800FAX 特価¥38,000
US Robotics COURIER V.34 TERBO 特価¥63,800

メーカー	型番	ツクモ特価
AIWA	PV-BF144	¥17,800
OMRON	ME1414B II (NEW)	¥19,800

通信ソフト

メーカー	製品	ツクモ特価
SPS	た～みのる2	¥13,000
SHARP	Communication SX-68K	¥15,800

ソフトウェア

製品	ツクモ特価
OS-9/X68030 V2.4.5	¥20,000
Technical Tool Kit V.2.4.5	¥16,000
Ultra C & Professional Pack V1.1	¥36,000
X Windows V11.5	¥24,000
SX-WINDOW Ver3.1システムキット	¥18,200
SX-WINDOWデスクアクセサリ集	¥11,800
C COMPILER Ver2.1 NEWKIT	¥35,800
Easydraw SX-68K	¥15,800
Easypaint SX-68K	¥10,200
SOUND SX-68K	¥12,600
Communication SX-68K	¥15,800
Matier Ver2.0	¥29,800
XL/Image	¥49,300
CD-ROM Driver	¥4,320
SX-PhotoGallery	¥14,220
DoubleBookin'	¥11,520
SX広辞苑(CD-ROM別)	¥17,800
EGWord SX-68K	¥47,800
SX-WINDOW開発キット	¥31,800
開発キット用ツール集	¥10,200
倉庫番リベンジSX-68K	¥5,400
MUSIC SX-68K	¥30,400
XDTP SX-68K	¥28,000
DataCalc SX-68K	¥47,800
フォント&ロゴデザインツール書家万流SX-68k	NOW WAITING
Super BUSINESS	NOW WAITING

★広告掲載価格は変動することもございます。お問い合わせください。

秋葉原

名古屋

札幌

お支払い方法

あなたのご都合に合わせていろいろ選べます。

クレジット払い

月々¥3,000以上の均等払いも頭金なし。夏・冬ボーナス2回払いもOK!

カード払い

¥5,000以上
通信販売での御利用カード
ツクモグローバルカード・セントラル・ジャックス
※御本人様より電話で通信販売部へお申し込み下さい。

各種リース払い

詳しくは各店にご相談下さい。

現金書留払い

〒101-91　東京都千代田区神田郵便局私書箱135号
ツクモ通販センター　Oh!X係

代金引き換え配達

お申し込みは電話1本でOK!
配達日の指定もできます。

銀行振込払い

BANK
事前にTELでお届け先をご連絡下さい。
三和銀行　秋葉原支店
(普)　1009939　ツクモデンキ

商品についてのお問い合わせは各店に

秋葉原

平日(営)AM10:45～PM7:30日・祝AM10:15～PM7:00

ツクモパソコン本店4F
03-3253-1899
03-3253-5599(代)
(休)木曜日

ツクモニューセンター店
03-3251-0987
(休)木曜日

名古屋

(営)AM10:30～PM7:30

ツクモ名古屋1号店
052-263-1655
(休)火曜日

ツクモ名古屋2号店
052-251-3399
(休)水曜日

札幌

平日(営)AM10:45～PM7:30日・祝AM10:15～PM7:00

ツクモ札幌店
011-241-2299
(休)木曜日

DEPOツクモ2番街店
011-242-3199
(休)木曜日

★商品はお電話受け付けより、標準日数3日～1週間でお届け致します。(一部地域を除く)
★表示価格には消費税は含まれておりません。

安いのに親切
TSUKUMO
九十九電機株式会社

マイコン専門ショップ

P&A

SHARPエキスパートショップ

今が購入のチャンス!

12/17〜1/17

X68030お買い得セット

(クレジット表：送料・消費税込み)

●CZ-500C
●CZ-608D(B)

定価￥492,800
P&A超特価
￥302,000

クレジット表	12回	27,400	24回	14,400
36回 10,000	48回	7,800	60回	6,500

●CZ-510C
●CZ-608D(B)

定価￥582,800
P&A超特価
￥401,000

クレジット表	12回	36,300	24回	19,100
36回 13,200	48回	10,300	60回	8,600

●CZ-300C
●CZ-608D(B)

定価￥482,800
P&A超特価
￥331,000

クレジット表	12回	30,000	24回	15,800
36回 10,900	48回	8,500	60回	7,100

●CZ-310C
●CZ-608D(B)

定価￥572,800
P&A超特価
￥396,000

クレジット表	12回	36,000	24回	18,900
36回 13,000	48回	10,200	60回	8,500

■モニター変更の場合
●CZ-615D(チューナー付)に変更の場合￥56,000
●CZ-621D(B)…………に変更の場合￥64,000
加算して下さい。

決算大処分セール 旧シリーズ今が買いどき!!

(送料￥2,000・消費税別)(クレジット表：送料・消費税込み)

X68000 Compact XVI

●CZ-674C-H
●CZ-608D(B)

定価￥392,800
P&A超特価**￥145,000**

12回	13,200	24回	7,000	36回	4,800	48回	3,800	60回	3,100

●CZ-674C-H
●CZ-608D(B)
●CZ-6FD5

定価￥492,600
P&A超特価**￥193,000**

12回	17,600	24回	9,200	36回	6,400	48回	5,000	60回	4,200

X68000 PRO II

●CZ-653C
●CZ-612D
(0.31mm、3モード TVチューナー チルト台付)

限定
定価￥404,800
P&A超特価**￥109,000**

X68000 PRO II-HD

(ハードディスク40MB内蔵)
●CZ-663C
●CZ-606D

定価￥474,800
限定
P&A超特価
￥118,000

単品(限定)

●CZ-652C
特価￥63,000
●CZ-653
特価￥65,000
●CZ-663C
特価￥75,000

MIDIセット

●MC-6600(SNE)
●SX-68M II(システムサコム)
●MIDIケーブル
特価￥48,500

●SC-55MK II(ローランド)
●SX-68M II(システムサコム)
●MIDIケーブル
特価￥70,800

(SC-88に変更の場合￥17,000加算して下さい。)

単品
●MC-6600(SNE)…………特価￥34,800
●SC-55MK II(ローランド)・特価￥56,800
●SC-88(ローランド)………特価￥73,500
●05R/W(KORG、加賀電子)・特価￥49,800

スピーカー

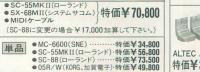
●MS-3000(SNE)………特価￥11,500
●SC-C55(AIWA)………特価￥ 5,980

ALTEC ACS300
特価￥37,000

ALTEC ACS100
特価￥16,000

ALTEC ACS50
特価￥9,200

X68000/68030用 メモリボード (送料￥700・消費税別)

■I/Oデータ
●SH-5BE4-8M(30用)…特価￥39,500
●SH-6BE1-1ME(600C用)…特価￥10,200
●PIO-6BE1-AE(ACE/PRO/PROII用)特価￥10,200
●PIO-6BE2-2ME(拡張スロット用)特価￥21,000
●PIO-6BE4-4ME(〃)・特価￥35,300

■シャープ
●CZ-5BE4(30用)………特価￥39,800
●CZ-5ME4(5BE4用増設)・特価￥36,500
●CZ-6BE2A(XVI用)……特価￥38,900
●CZ-6BE2B(XVI、674C増設)特価￥37,500
●CZ-6BE2D(674C用)…特価￥20,500

モデム&FAXモデム (送料￥1,000)

〈アイワ〉
●PV-BF144(ボックス型)
…………………………特価￥17,000
●PV-AFV144(液晶パネル、ボックス型)
…………………………特価￥26,800
●PV-PFV144(ポケット型)
…………………………特価￥22,800

〈オムロン〉
●ME1414B II(ボックス型)………特価￥17,000
●MD144XT10V(ボックス型)……特価￥34,000

〈マイクロコア〉
●MC14400FX(W)(ボックス型)…特価￥23,000
●MC24FC5(W)(ポケット型)……特価￥20,000

注目!!平成7年3月末払い手数料(金利)無料(平成7年1月末/2月末/3月末のいずれかをご指定下さい。)

MO&CD-ROM (送料￥1,000)

■CS-M230PA(コパル)
光磁気ディスク(X68000用)
●ケーブル付
特価￥102,000

■LMO-FMX330TS
(ロジテック)●ケーブル付
定価￥168,000
特価￥97,000

■MO (ケーブル別売)
●UL-312E-S(緑電子)………特価￥62,000
●MO-120S(ICM)……………特価￥88,000
●MO-230S(〃)……………特価￥110,000
●LMO-340(ロジテック)………特価￥85,000
●LMO-400(〃)………特価￥110,000

■CD-ROM (ケーブル別売)
⊙SCD-200(ロジテック) P&A特価￥17,000
※CD-ROM Driver+SCSIケーブル・特価￥7,300

東京システムリサーチ製(X SIMM)

(送料￥700・消費税別)

(X SIMM VI)
⊙X VIシリーズ専用SIMM増設式メモリボード
●X SIMM VI(634C用)…定価￥16,500➡特価￥13,000
●X SIMM VIc(674C用)・定価￥16,500➡特価￥13,000
⊙増設SIMMメモリ(72PIN)
●4MB(70ns)……………………特価￥11,800
●8MB(70ns)……………………特価￥27,800
●4MB(60ns、24MHz以上用)………特価￥16,500
●8MB(60ns、24MHz以上用)………特価￥28,000

●6MB(70ns、メーカー純正品)……特価￥27,800
(X SIMM 10) ⊙SIMM増設式メモリボード
●X SIMM 10………定価￥18,000➡特価￥15,700
⊙増設SIMMメモリ ●1MB×2……特価￥ 9,000
●4MB×2……特価￥30,000
●10MB例 X SIMM10+1MB×2+4MB×2‥￥54,700

X68000/68030専用ハードディスク (送料￥1,000・消費税別)

外付

■ロジテック
⊙SHD-B340NU(340MB、12ms)
(ケーブル、ターミネータ付)…特価￥35,800
⊙SHD-B540U(540MB、10.5ms、256K)
(ケーブル、ターミネータ付)…特価￥49,800

■モッキンバード
⊙HD-M350(350MB、14ms、256K)
…………………………特価￥35,800
⊙HD-K520(520MB、12ms、240K)
…………………………特価￥48,800
⊙HD-M1000(1070MB、9.5ms、512K)
…………………………特価￥92,800

内蔵

■CZ-500C/300C専用
⊙CZ-5H08(80MB/23ms)
定価￥ 98,000▶特価￥71,800
⊙CZ-5H16(160MB/18ms)
定価￥135,000▶特価￥99,500

●価格は変動します。ご注文の際は必ずお電話で価格と在庫をご確認下さい。●本広告に掲載の商品には送料及び消費税は含まれておりません。

ズバリご奉仕

P&Aならではの 新品パソコン5年保証

《業界№1の"P&Aメンテナンスサポート"》

最高の保証システム

①業界最長の新品パソコン5年保証
（※モニター・プリンター3年間保証!! ※一部商品は除きます。）
②中古パソコンの1年間保証（※モニター・プリンター6ヶ月間保証!!）
③初期不良交換期間3ヶ月（※新品商品に限らせていただきます。）
④永久買取保証
⑤配達日の指定OK!!（土曜・日曜・祭日もOK!!）
⑥夜間配達もOK!!（※PM6:00〜PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム

①翌月一括払い手数料無料（ご利用下さい。）
②業界№1の低金利!!
③月々の支払いは￥1,000より
④9ヶ月先からのスキップ払いOK!!
⑤84回までの分割、ボーナス併用OK!!
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK!!
⑨現金一括支払いOK!!
⑩商品到着払いOK!!（代引き手数料が必要になります。10万円まで900円）
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

●法人向け リースシステム
業務に最適なシステムを構築します。
損金処理が可能なリース契約をどうぞ。

周辺機器コーナー （送料￥1,000・消費税別）

カラーイメージスキャナ
■JX-330X
定価￥178,000
特価**￥118,000**

カラーイメージジェット

■IO-735X-B
定価￥248,000
特価**￥128,000**

ビデオスキャナー
■CZ-6VS1
定価￥178,000
特価**￥135,000**

FDD（5インチ×2基）
■CZ-6FD5
定価￥99,800
P&A超特価
￥49,800

プリンター（ケーブル用紙付）

- MJ-500V2（エプソン）……特価￥31,300
- MJ-1000V2（〃）……特価￥51,300
- MJ-700V2C（〃）……特価￥65,300
- BJ-220JCII（キヤノン）…特価￥53,400
- BJ-10V Lite（〃）……特価￥31,300
- BJ-15V PRO（〃）……特価￥39,700
- LBP-A404GII（〃）……特価￥87,300
- BJC-600J（〃）……特価￥71,300
- BJC-400J（〃）……特価￥54,300

ペン&タブレット

■Drawing Slate（NS CalComp）
●31090SER
定価￥74,800
特価**￥58,500**

- CZ-6BV1………定価￥21,000▶特価￥15,900
- CZ-8NM3………定価￥ 9,800▶特価￥ 7,200
- SH-6BF1………定価￥49,800▶特価￥36,500
- CZ-6BP1………定価￥79,800▶特価￥57,000
- CZ-6BS1………定価￥29,800▶特価￥21,500
- CZ-8NJ2（限定）…定価￥23,800▶特価￥13,800
- CZ-6CS1（674C用）・定価￥12,000▶特価￥ 8,900
- CZ-6CR1（RGBケーブル）・定価￥ 4,500▶特価￥ 3,600
- CZ6CT1（テレビコントロール）・定価￥ 5,500▶特価￥ 4,400
- CZ-6BP2………定価￥45,800▶特価￥33,300
- CZ-5MP1（X68030用）・定価￥54,800▶特価￥42,000

送料￥700・消費税別

■システムサコムボード
●SX-68MII（MIDI）
定価￥19,800
特価￥13,500
●SX-68SC（SCSI）
定価￥26,800
特価￥17,500

X68000用ソフトコーナー （送料￥700・消費税別）

〈シャープ〉
- CYBERNOTE PRO68K（CZ-243BSD）……特価￥15,000
- MUSIC PRO68K（MIDI）（CZ-247MSD）……特価￥20,500
- CANVAS PRO68K（CZ-249GSD）特価￥22,000
- Communication PRO68K Ver.2.0（CZ-257CSD）……特価￥15,300
- Easypaint SX-68K（CZ-263GWD）……特価￥ 9,800
- Easy draw SX-68K（CZ-264GWD）・特価￥15,300
- New Print Shop Ver.2.0（CZ-265HSD）……特価￥15,400
- Press Conductor PRO68K（CZ-266BSD）……特価￥22,000
- CHART PRO68K（CZ-267BSD）…特価￥29,800
- EG-Word（CZ-271BWD）……特価￥44,900
- Communication SX68K（CZ-272CWD）……特価￥14,500
- Datacalc SX-68K（CZ-273BWD）……特価￥44,000
- MUSIC SX68K（CZ-274MWD）…特価￥29,300
- SOUND SX68K（CZ-275MWD）…特価￥11,500
- フォント・アンド・ロゴデザインツール SX-68K（CZ-282BWD）……特価￥22,000
- BUSINESS PRO68K（CZ-286BSD）……特価￥20,500
- 開発キット（work room）（CZ-288LWD）……特価￥29,700
- 開発キット用ツール集（CZ-289TWD）……特価￥ 9,600
- SX-WINDOWディスクアクセサリー集（CZ-290TWD）……特価￥11,500
- XOTP-SX68K（CZ-291BWD）……特価￥26,900
- C-Compiler PRO68K Ver.2.1（CZ-295LSD）
- NEW KIT……特価￥32,500
- SX-WINDOWS Ver.3.1（CZ-296SS/SSC）……特価￥17,600
- 倉庫番リベンジ SX-68K（CZ-293AW/AWC）……特価￥ 5,100

〈マイクロウェア〉
- OS-9/X68030 V.2.4.5……特価￥19,900
- X-WINDOWS V.11 R5……特価￥25,500
- Technical Tool Kit V.2.4.5……特価￥17,000
- Ultra C アンド Professional Pack V.1.1……特価￥38,000
- Video PC for X680X0……特価￥57,000

〈計測技研〉
- SX広辞苑（CD-ROMバンドル版）……特価￥37,000
- Double Bookin……特価￥ 9,600
- CD-ROM Driver V.2.0……特価￥ 3,800
- SX-Photo Gallery……特価￥13,400

〈その他〉
- F-Card V5 for X68K（クレスト）……特価￥ 9,600
- F-Calc for X68K（クレスト）……特価￥11,000
- たーみのる2（SPS）……特価￥13,000
- MU-1GS（サンワード）……特価￥21,000
- マチエール Ver.2.0（サンワード）……特価￥28,800
- Z's STAFF PRO68K Ver.3.0（ツァイト）……特価￥37,500
- Z's TRIPHONY デジタルクラフト（ツァイト）……特価￥27,000

全国通販

★頭金なし!
★即日発送

●お近くの方はお立寄り下さい。専門係員が説明いたします。
●本体単品で特価で受付します。詳しくは電話にてお問合せ下さい。
●ビジネスソフト定価の20%引きOK! TELください。

P&A特選 今月の中古特選品

●CZ-674C
●68000専用モニター付
￥96,000

●CZ-623C
●68000専用モニター付
￥96,000

●CZ-653C
●68000専用モニター付
￥79,000

新品限定
- CZ-652C ……￥63,000
- CZ-653C ……￥65,000
- CZ-663C ……￥75,000

- CZ-600C…￥40,000
- CZ-601C…￥40,000
- CZ-611C…￥45,000
- CZ-652C…￥50,000
- CZ-612C…￥60,000
- CZ-603C…￥53,000
- CZ-653C…￥55,000
- CZ-612C…￥65,000
- CZ-623C…￥75,000
- CZ-674C…￥73,000
- CZ-634C…￥110,000
- CZ-644C…￥145,000

※上記は単品価格、モニター別売。

高額買取り（新品もOK）格安販売

■まずはお電話下さい。
下取り専用 買取り電話▶ ☎03-3651-1884 FAX.03-3651-0141

買取り価格…完動品・箱／マニュアル／付属品の価格です。中古販売…1年間保証付。

●下取りの場合…価格は常に変動していますので査定額を電話で確認してください。（差額は、P&A超低金利クレジットをご利用ください。）
●買取りの場合…現品が着き次第、3日以内に高価買取金額を連絡し、振込み、又は書留でお送り致します。

●最新の在庫情報・価格はお電話にてお問い合せください。
●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せください。
●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認ください。
●本商品の掲載の商品の価格については、消費税は、含まれておりません。
●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せください。

P&Aオリジナル特選パソコンラック&OAチェアー （消費税込み）（送料無料、離島を除く）

①**￥10,815**（2段別々使用OK）

※キャスター付、4段、17"モニターOK、色（グレー）。
※上から2番目棚板移動可能。

②**￥12,360**（マウステーブルスライドOK）

※キャスター付、4段、17"モニターOK、色（グレー）。
※スライドマウステーブル、中棚板は2段階移動可能。

①**￥4,944**
●布張り 色（グレー）
●ガス圧シリンダー

②**￥6,283**
●肘付
●布張り 色（グレー）
●ガス圧シリンダー

※ラック、チェアー持ち帰り可能です。ご来店下さい。

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。●1回〜84回払いまで出来ます。但し、1回のお支払い額は￥1,000円以上。

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。（電信扱いでお振込み下さい。）

〔振込先〕さくら銀行 新小岩支店
当座預金 2408626 （株）ピー・アンド・エー

超低金利クレジット率

回数	3	6	10	12	15	24	36	48	60	72
手数料	2.6	3.5	4.4	4.9	7.8	10.4	14.4	18.9	24.4	31.8

※お支払いは、便利な商品到着払い〈手数料（10万円まで900円）要〉をご利用下さい。

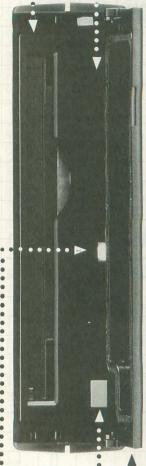
P&A 株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目2番地20号
●営業時間：AM10：00〜PM7：00 日・祭：AM10：00〜PM6：00
☎03-3651-0148（代）
●定休日／毎週水曜日 FAX.03-3651-0141 MAC/DOS Vフロア ☎03-3655-4454

Multimedia Supporter

COPAL

マルチプラットフォーム対応 MOディスクユニットの原寸広告です。

ディスク挿入口
光ディスクカートリッジの挿入/排出口です。

通風口
内蔵ファンを持たない無音設計です。

アクセス表示
光ディスクがデータにアクセスしている際に点灯します。

イジェクトスイッチ
光ディスクカートリッジを取り出します。

フロントカバー
携帯時にはカバーを閉じれば防塵は万全。

超小型・超軽量MOドライブ、デビュー。

■A5サイズより小さい世界最小クラスのMO。
ビデオカセットほどのコンパクトなボディは、わずか695gの軽さ。ノートパソコンと一緒に携帯できる抜群のポータビリティと、デスクトップでの省スペースを実現しました。

■各種のパソコンに接続できるマルチプラットフォーム対応。
モードスイッチを切替えるだけで、IBM PC、AT互換機、DOS/V、PC98、Mac、Power Macなどの各種パソコンに対応するマルチプラットフォーム。オプションのドライバソフトを使用すれば異機種間でのデータ交換も可能です。(MS-DOS機)

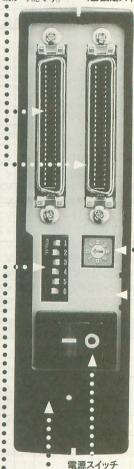
■大容量230MBはもちろん128MB媒体も使用可能。
国際標準ECMA201準拠の光磁気ディスクカートリッジは、1枚でフロッピーディスク180枚分とハードディスク並みの大容量。さらにISO準拠の128MBカートリッジもそのまま使用でき、今までの資産を有効に活用できます。

■SCSI-2準拠の高性能・高速ドライブ採用。
SCSI-2準拠の高速ドライブ採用により、同期転送時5MB/sの高速動作を実現。平均シークタイム40ms以下、回転数3,600rpmで、高速データアクセスを提供します。

PowerMO™
3.5 inch MO Disk Unit
230MB

SCSIコネクタ
2つのSCSIコネクタ(ハーフピッチ)で、ディジーチェーン接続が可能です。

ID設定スイッ

電源スイッチ

電源ジャック
電源アダプタのプラグを差込みます。

エアフィルタ
不要

モードスイッチ
接続するパソコンやインタフェースごとに使うモードの設定、アクティブターミネータのON/OFF設定をします。

PowerMO™ PM230A

主な仕様 ●230MB/ECMA201準拠3.5インチ光磁気ディスクカートリッジ(片面)、128MB/ISO準拠3.5インチ光磁気ディスクカートリッジ(片面)●バイト数、512B/セクタ●ホストインタフェース/SCSI-2●回転数/3600rpm●平均シーク時間/40ms以下●バッファ容量/243KB●ホスト転送レート/5.0MB/s(同期)、2.5MB/s(非同期)●外形寸法/W117×D209×H30●消費電力/8W(最大)●重量695g(本体重量)●カラー/ブラック、グレー、ライトグレー

標準価格 ¥148,000(ACアダプタ付) ※記載の会社名および製品名は各社の商標または登録商標です。

マルチメディア・サポーター
株式会社コパル
〒174 東京都板橋区志村2丁目16番20号
■製品のお問い合わせ先:ユーザーサポート係
TEL:03-3965-1161 FAX:03-3558-9455
(月~金 9:30~12:00 13:00~17:00 土日祝祭日を除く)

T1002179010681 雑誌 02179-1